# NUMERICAL ELECTROMAGNETIC CODE (NEC) - BASIC SCATTERING CODE

## PART II: CODE MANUAL

F. W. Schmidt and R. J. Marhefka

The Ohio State University

ElectroScience Laboratory

Department of Electrical Engineering
Columbus, Ohio 43212

Technical Report 784508-14

September 1979

Naval Regional Procurement Office
Long Beach, California 90822

Approved for public release;
distribution unlimited.

# NOTICES

When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely related Government procurement operation, the United States Government thereby incurs no responsibility nor any obligation whatsoever, and the fact that the Government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication or otherwise as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER

2. GOVT ACCESSION NO. AD-A097 417

3. RECIPIENT'S CATALOG NUMBER

4. TITLE (and Subtitle)
NUMERICAL ELECTROMAGNETIC CODE (NEC) -
BASIC SCATTERING CODE -
PART II: CODE MANUAL

5. TYPE OF REPORT & PERIOD COVERED
Technical Report

6. PERFORMING ORG. REPORT NUMBER
ESL-784508-14

7. AUTHOR(s)
F. W. Schmidt ~~and~~ R. J. Marhefka

8. CONTRACT OR GRANT NUMBER(s)
15
Contract N00123-76-C-1371

9. PERFORMING ORGANIZATION NAME AND ADDRESS
The Ohio State University ElectroScience
Laboratory, Department of Electrical Engineering
Columbus, Ohio 43212

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS

11. CONTROLLING OFFICE NAME AND ADDRESS
Naval Regional Procurement Office
Long Beach, California 90822

12. REPORT DATE
11 September 1979

13. NUMBER OF PAGES
480

14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)

15. SECURITY CLASS. (of this report)
Unclassified

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES
The material contained in this report is also used as a thesis submitted to the Department of Electrical Engineering, The Ohio State University as partial fulfillment for the degree Master of Science.

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

| | |
|---|---|
| Geometrical Theory of Diffraction | Cylinder model |
| Uniform asymptotic solutions | Far field pattern |
| Plate model; | Computer code |

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The Numerical Electromagnetic Code - Basic Scattering Code is a user-oriented computed code for the analysis of the far field patterns of antennas in the presence of perfectly conducting metal structures at UHF and above. The analysis is based on uniform asymptotic techniques formulated in terms of the Geometrical Theory of Diffraction (GTD). Complicated structures can be simulated by arbitrarily oriented flat plates, an infinite ground plane, and a finite elliptic cylinder.

20.

A wide range of practical problems can be simulated using these shapes. For example, flat plates can be used to model the superstructure of a ship, the body of a truck, or the wings and stores of an aircraft. The finite elliptic cylinder can be used to model a mast or smoke stack of a ship, or the fuselage and engines of an aircraft.

This document describes the FORTRAN coding in detail. It gives background on practical aspects of the GTD and contains an overview of the code organization. This information will be of primary interest to someone attempting to modify the code. It will also be helpful when the code is being implemented on a computer system on which the coding may not be compatible.

## TABLE OF CONTENTS

# CHAPTER I
# INTRODUCTION

The Numerical Electromagnetic Code - Basic Scattering Code is a user-oriented computer code for the analysis of the far field patterns of antennas in the presence of perfectly conducting metal structures at UHF and above. Complicated structures can be simulated by arbitrarily oriented flat plates, an infinite ground plane, and a finite elliptic cylinder. This type of analysis has been used very successfully in the past to model aircraft shapes[1,2,3]. The present solution has been extended to include a wide range of problems. For example, flat plates can be used to model the superstructure of a ship, the body of a truck, or the wings and stores of an aircraft. The finite elliptic cylinder can be used to model a mast or smoke stack of a ship, or the fuselage and engines of an aircraft.

The analysis is based on uniform asymptotic techniques formulated in terms of the Geometrical Theory of Diffraction (GTD)[4,5,6]. The GTD approach is ideal for a general high frequency study of antennas in a complex environment in that only the most basic structural features of an otherwise very complicated structure need to be modeled. This is because ray optical techniques are used to determine components of the field incident on and diffracted by the various structures. Components of the diffracted fields are found using the GTD solutions in terms of the individual rays which are summed with the geometrical optics terms in the far field. The rays from a given scatterer tend to interact with other structures causing various higher-order terms. In this way one can trace out the various possible combinations of rays that interact between scatterers and determine and include only the dominant terms. Thus, one need only be concerned with the important scattering components and neglect all other higher-order terms. This method leads to accurate and efficient computer codes that can be systematically written and tested. Complex problems can be built up from simpler problems in manageable pieces.

The limitations associated with the computer code result from the basic nature of the analyses. The solution is derived using the GTD which is a high frequency approach. In terms of the scattering from plate structures this means that each plate should have edges at least a wavelength long. In terms of the cylinder structure its major and minor radii and length should be a wavelength in extent. In addition, each antenna element should be at least a wavelength from all edges and the curved surface. In many cases, the wavelength limit can be reduced to a quarter wavelength for engineering purposes.

Modeling small structures and antennas can be better accomplished using an integral equation solution such as NEC-Moment Methods[7]. The Basic Scattering Code has been interfaced with the Moment Method code so that the capabilities of both methods can be used to the fullest. For example, the Moment Method code can be used to analyze the currents and impedance of an antenna. The magnitude and phase of the current weights can then be used in the Basic Scattering Code to predict the far field patterns of the antennas in arbitrary pattern cuts.

There are two documents describing the NEC-Basic Scattering Code. Part I is a User's Manual[8] that contains a detailed description of the input parameters, an interpretation of the output, and example problems. The example problems are composed of sample input data with the resulting far field patterns compared against known results to confirm the validity of the code. Most users of the code will find that the User's Manual is sufficient to learn how to effectively operate the code.

This document is Part II. It describes the FORTRAN coding in detail. Chapter II gives background on practical aspects of the GTD. Several examples are shown to illustrate how the various GTD fields superimpose to give a total solution. Next, a particular GTD term is discussed in more detail to show the general concepts involved throughout the code. Chapter III contains an overview on how the code is organized. It describes the various coordinate systems involved, how a general subroutine is organized, and how the various subroutines are interrelated. Chapter IV contains for each subroutine: (1) a statement of purpose, (2) an illustration showing the geometry involved, (3) a brief narrative on the method used, (4) a flow diagram, (5) a dictionary of major variables, (6) a listing of the code. Chapter V defines the common blocks and Chapter VI lists the system library functions used by the code.

The information in the Code Manual will be of primary interest to someone attempting to modify the code. It will also be helpful when the code is being implemented on a computer system on which the coding may not be compatible.

## CHAPTER II
## BACKGROUND

The Basic Scattering Code is used to evaluate the far field patterns of a given antenna in the presence of perfectly conducting scattering structures. It is a useful tool in the analysis and design of antenna placement and performance. This section provides the reader with background on how GTD is used in the code for computing the scattered fields. It also shows how to interpret and correlate the computed scattered fields to the specific geometry of a scattering structure. This chapter also provides a simple view of how the code generates a specific GTD scattered term. The explanations provided are general, giving an introduciton to the more detailed descriptions provided later in the code manual. For a theoretical anlaysis of the methods used in the code, the reader is encouraged to refer to References 1, 4, 5, 6.

### A. Qualitative Overview of GTD

The goal of the code is to solve for the fields scattered in a specified direction from a source (or set of sources) by the various features of a structure, as shown in Figure 1. The total field in a given observation direction is obtained by taking the sum of fields resulting from a number of different scattering mechanisms. Each component is determined by tracing the ray through the appropriate geometrical path and then using the Uniform Geometrical Theory of Diffraction to compute the magnitude and phase of the field if it has not been shadowed. The following examples serve to show the different mechanisms used in computing the scattered field and an example of typical fields resulting from such mechanisms.

#### Example consisting of a source and a single scattering element

The geometry used is a half-wave electric dipole mounted two wavelengths above a square plate four wavelengths on a side as shown in Figure 2. The total field of the source and structure is given by, $\bar{E} = \bar{E}^i + \bar{E}^s$ where $\bar{E}^i$ is the incident field:

$$\bar{E}^i = \begin{cases} \text{incident source field,} & \text{where ray is not shadowed} \\ 0 \quad , & \text{where source ray is shadowed,} \end{cases}$$

and $\bar{E}^s$ is the GTD scattered field:

$$\bar{E}^s = \begin{cases} \text{scattered fields,} & \text{where the rays are not shadowed,} \\ 0 \quad , & \text{where the rays are shadowed.} \end{cases}$$

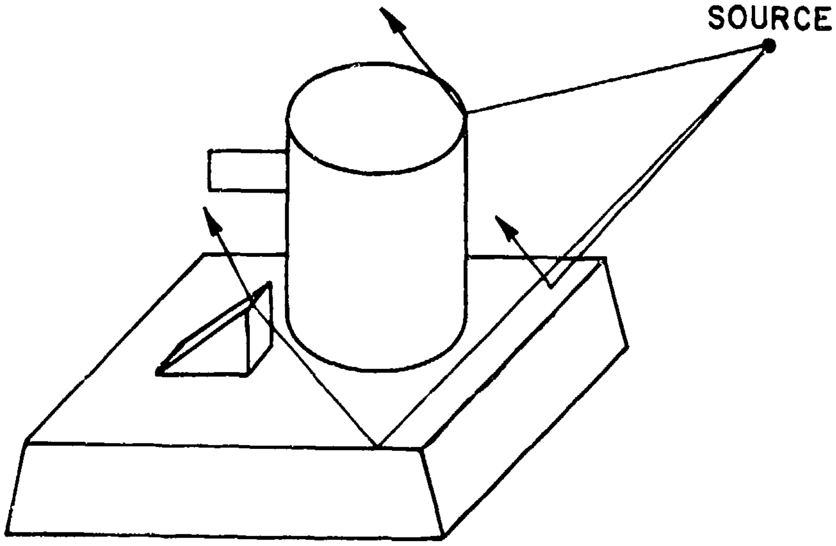


Figure 1--Illustration of general GTD problem.

The GTD scattered field is composed of the reflected fields, diffracted fields, etc. The source and reflected fields comprise the geometrical optics fields (G.O.).

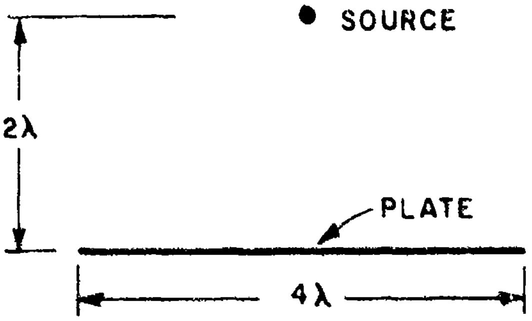


Figure 2--Geometry for a source in the presence of a plate.

Several single-order terms are used to compute the fields (in the far-zone) scattered by this structure. The word "order" here refers to the number of times the particular scattering term interacts with the body as it propagates from the source to the observation point. The source (or incident) field is that field which propagates straight from the source into the far field in the direction of the observer as shown in Figure 3. The pattern of the source field, in the presence of the plate, taken in the plane of the page is shown in Figure 4. The scale used in the patterns of Figures 4-10 is 0 to -40 dB. They are normalized to the maximum value of the total field pattern in Figure 10.

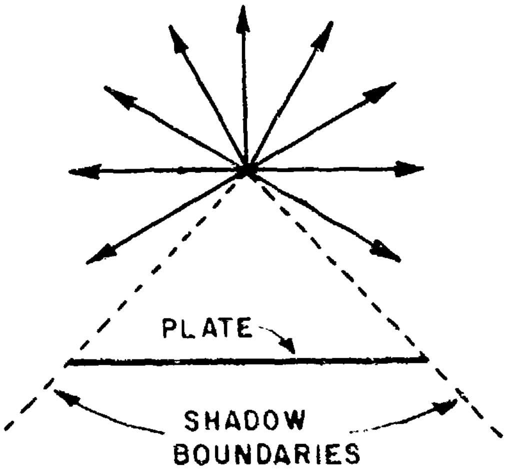


Figure 3--Illustration of source ray paths.

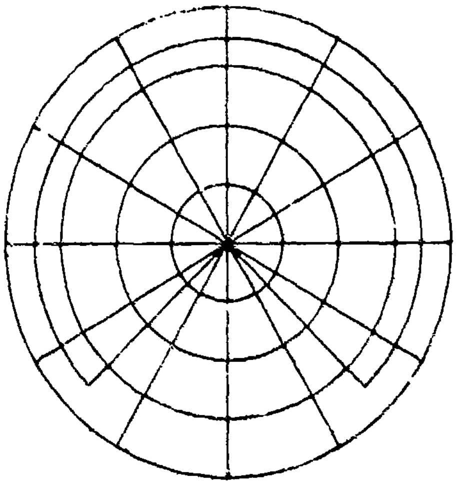


Figure 4--Source field.

The reflected field is simply the geometrical optics field reflected by the plate as shown in Figure 5. The fields due to the reflection mechanism, shown in Figure 6, are easily obtained from image theory. The total geometrical optics fields (the sum of the direct and reflected fields) are plotted in Figure 7.

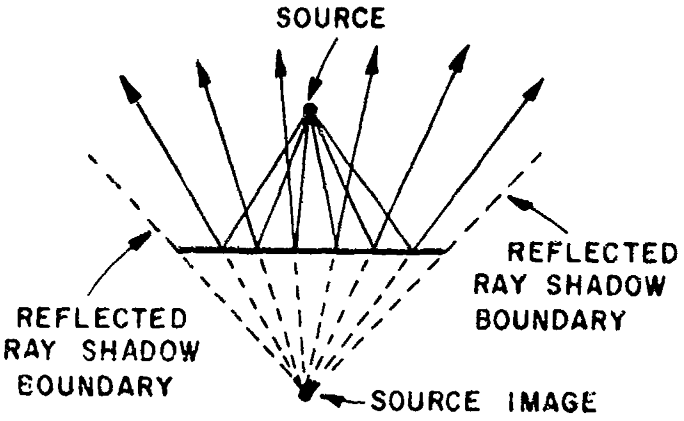


Figure 5--Illustration of plate reflected ray paths.

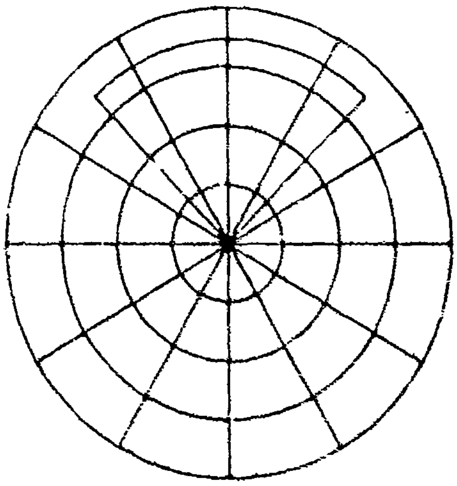


Figure 6--Field reflected from plate.

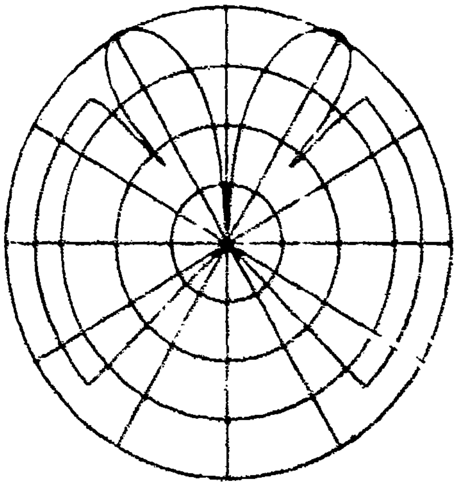


Figure 7--Geometrical optics field which is the sum of the incident and reflected fields.

The diffracted fields, which include edge, slope, and corner diffraction, are shown in Figure 8. The ray paths for edge diffraction are shown in Figure 9. Figure 10 shows the total scattered field. Note that the diffracted field smoothes out the discontinuities on the G.O. fields. Although the diffracted field magnitude is continuous at the shadow boundary, the phase jumps by 180° there. This subtracts from the lite side and adds to the shadow side, smoothing out the discontinuity. Higher order terms (such as double diffraction) could also be computed to further improve the accuracy of the solution.

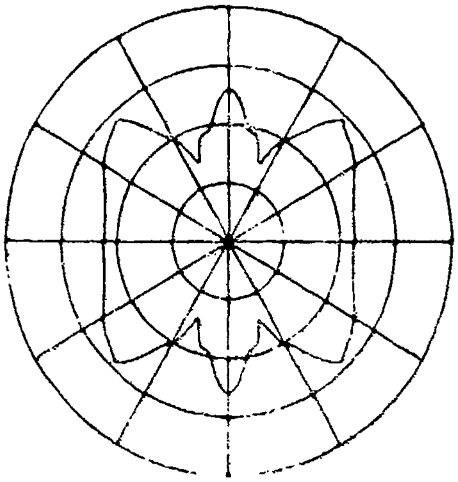


Figure 8--Diffracted fields.

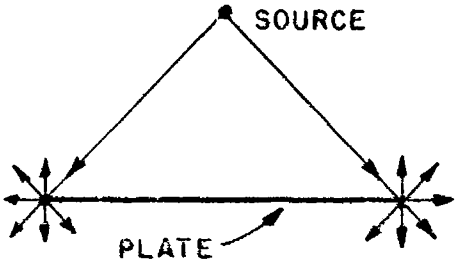


Figure 9--Illustration of diffracted rays.

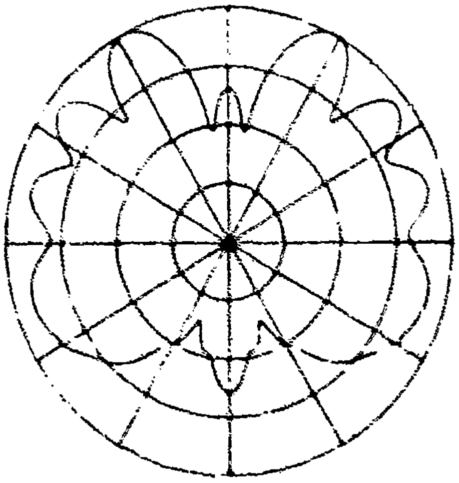


Figure 10--Total pattern.

Example consisting of a source and three scattering elements

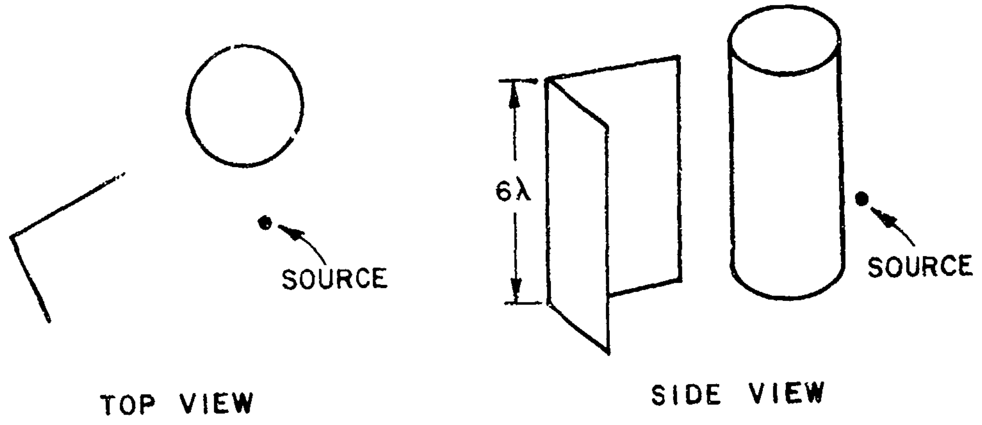


Figure 11--Illustration of source and scattering elements.

The geometry used for this example is shown in Figure 11. As with the previous example, the source field, single order plate reflections, and diffractions exist, as is shown in Figures 12-16. Field patterns in Figures 11-36 are taken in the plane normal to the cylinder and plotted with a scale from 0 to -40 dB such that they are normalized to the maximum value of the total field pattern in Figure 36.

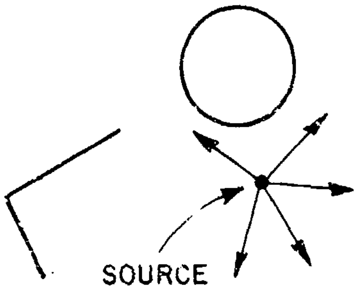


Figure 12--Source field ray paths.

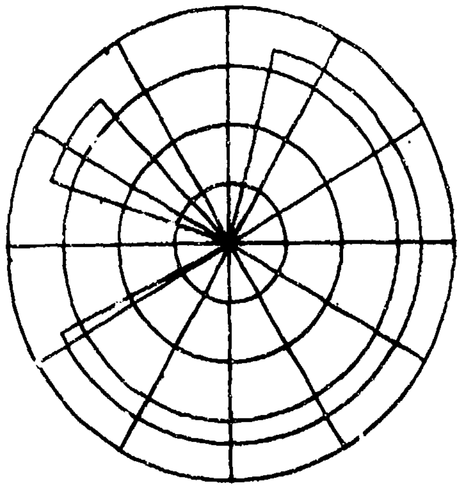


Figure 13--Source fields.

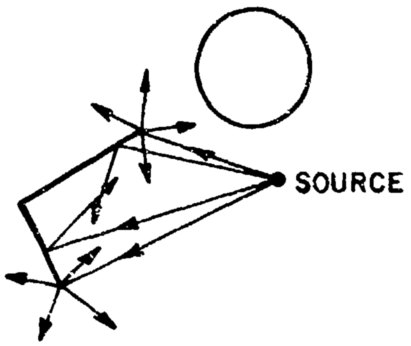


Figure 14--Illustration of first order plate ray paths.

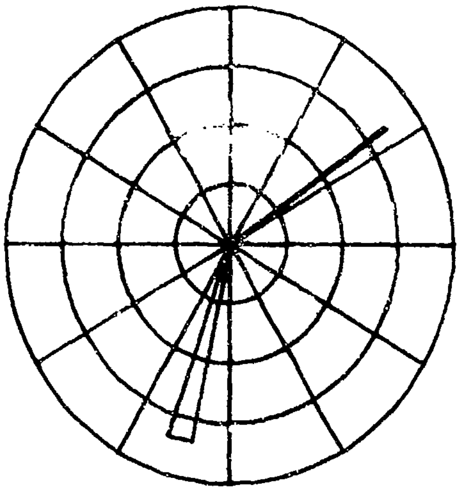


Figure 15--Fields due to single order plate reflection.

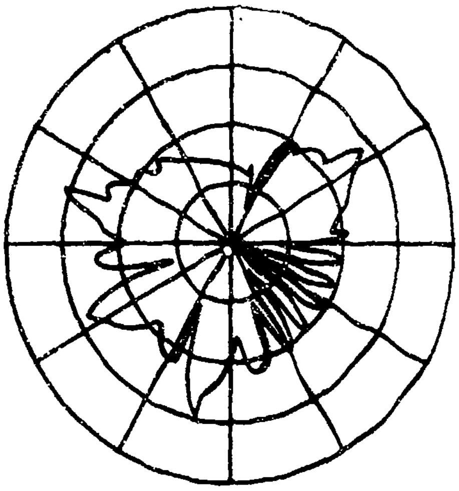


Figure 16--Fields due to plate diffraction.

In addition to first order plate terms there are first order cylinder terms: 1) the scattered (reflected and diffracted) fields from the cylinder's curved surface, 2) the field reflected from the end caps and 3) the fields diffracted by the end cap rims. These are shown in Figures 17-21. Note that in the geometry presented in Figure 11, end cap reflections will not occur. Therefore, a different geometry is shown in Figure 21 to demonstrate end cap reflections. Note that with more than one body present, individual terms are often shadowed by other bodies in the structure, creating discontinuities as shown in many of the figures (as in Figure 18 for the cylinder scattered fields).

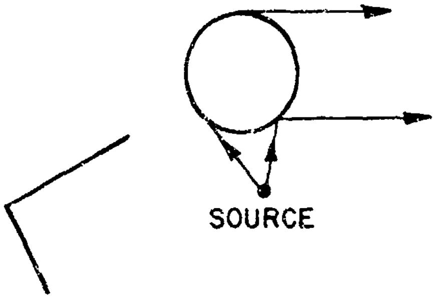


Figure 17--First order ray paths for the cylinder's curved surface.

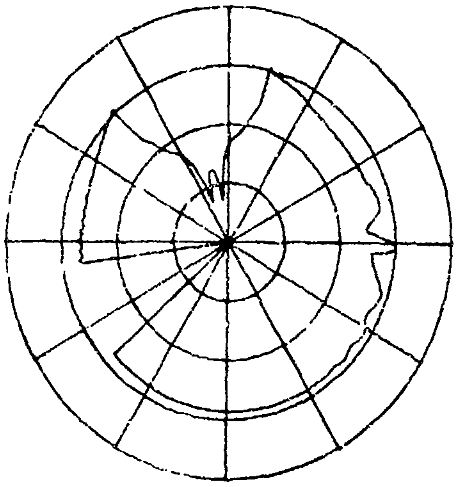


Figure 18--First order cylinder curved surface scattered field.

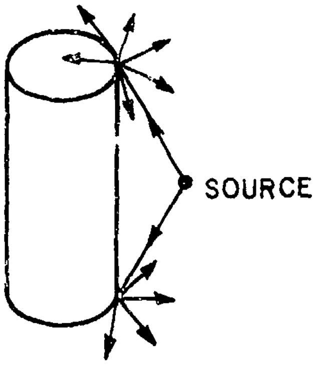


Figure 19--Ray paths for end cap diffracted fields.

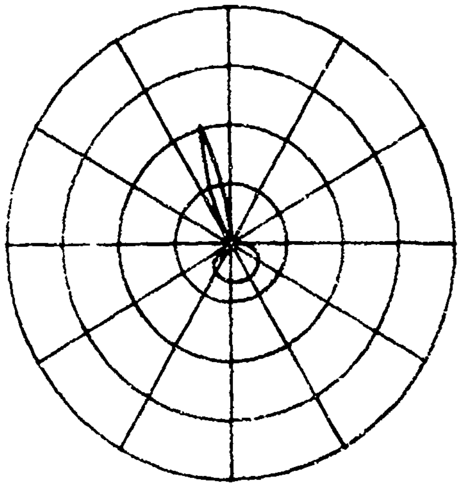


Figure 20--Fields due to end cap diffraction.

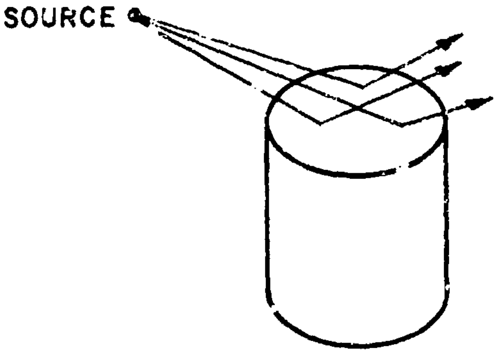


Figure 21--Illustracion of ray paths for end cap reflected fields.

In addition to single order mechanisms, second order scattering occurs where the ray is scattered by one body and then scattered by the second. Several different double scattering (or second order) terms are computed. Double reflection, where a ray is reflected by one plate and then by another, is shown in Figures 22 and 23.

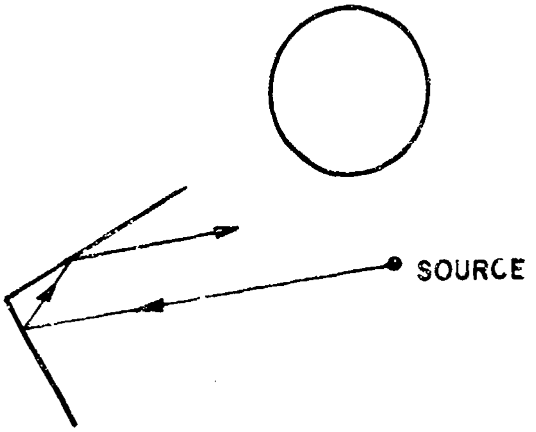


Figure 22--Ray path for double reflected fields.

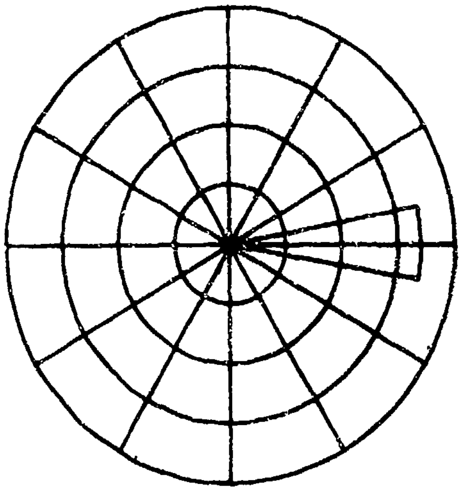


Figure 23--Fields due to double reflected rays.

Another second order scattering mechanism involving plates is reflection-diffraction, where a ray is reflected from one plate and diffracted by an^ther. This is illustrated in Figures 24 and 25. The inverse mechanism, diffraction-reflection, illustrated in Figures 26 and 27, involves fields diffracted from a plate edge and then reflected by another plate.

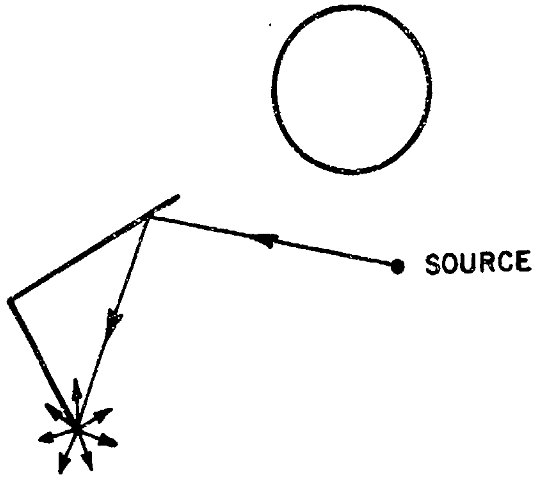


Figure 24--Ray paths for plate reflection-diffraction.

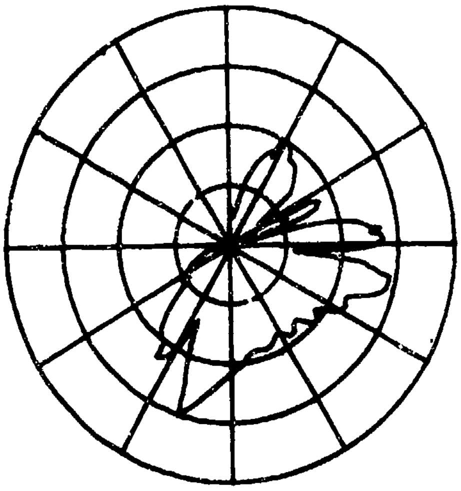


Figure 25--Fields resulting from plate reflection-diffraction.

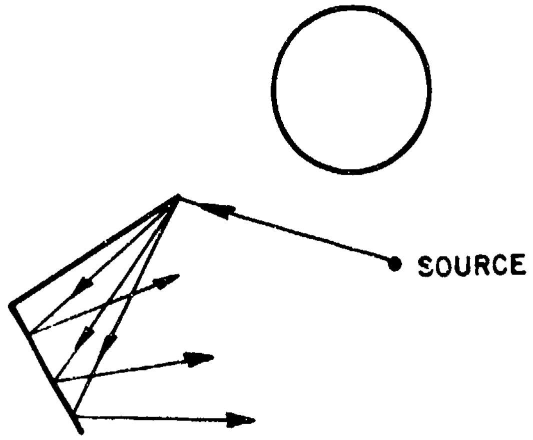


Figure 26--Illustration of plate diffracted-reflected ray paths.

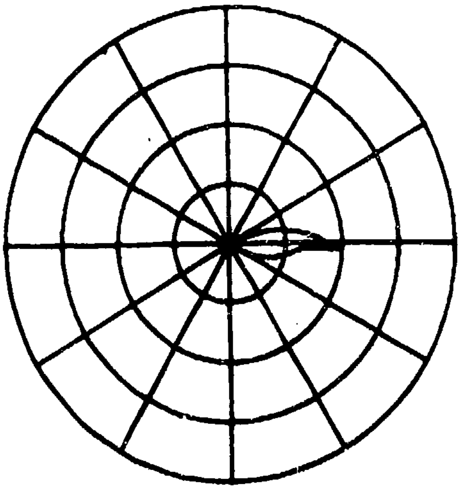


Figure 27--Fields due to plate diffraction-reflection mechanism.

A number of the scattering mechanisms involve interactions between the cylinders and one of the plates. Two such terms result from scattering of the fields by the cylinder and then reflection by a plate and vice-versa. Figures 28 and 29 illustrate the ray paths and fields of rays which are reflected from a plate and then scattered by the elliptic cylinder. Figures 30 and 31 illustrate ray paths and fields resulting from ray scattered by the cylinder and then reflected by a plate.

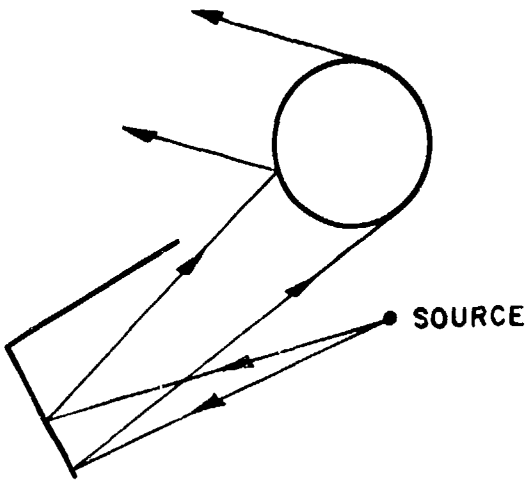


Figure 28--Illustration of rays reflected by a plate and scattered by the cylinder.

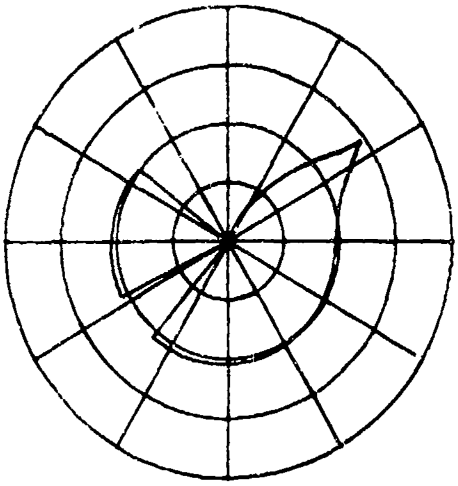


Figure 29--Fields reflected by plates and then scattered by the cylinder.

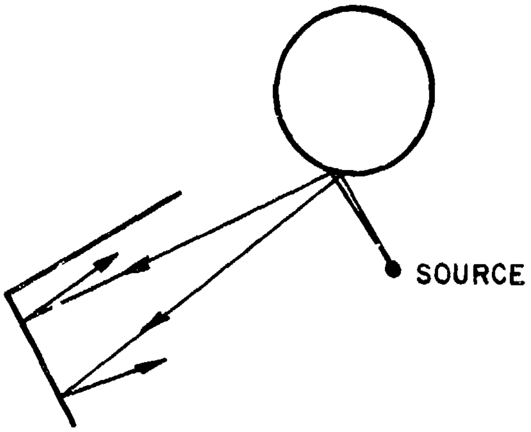


Figure 30--Illustration of rays scattered by cylinder and then reflected by a plate.

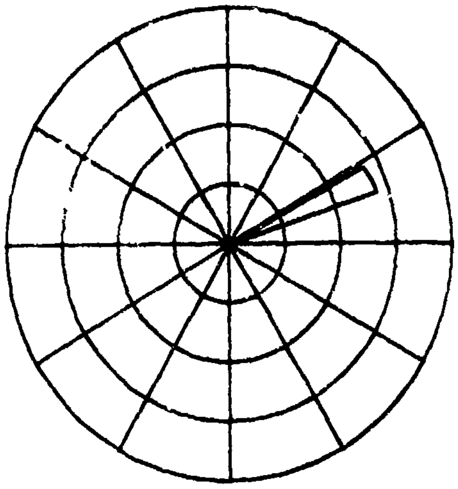


Figure 31--Fields scattered by the cylinder and reflected by a plate.

Another second order scattering mechanism involves fields reflected by the cylinder and then diffracted by a plate edge. The ray paths and fields for this term are illustrated in Figures 32 and 33. The inverse of this term is the fields of rays diffracted by a plate edge and then reflected by the cylinder, as shown in Figures 34 and 35.

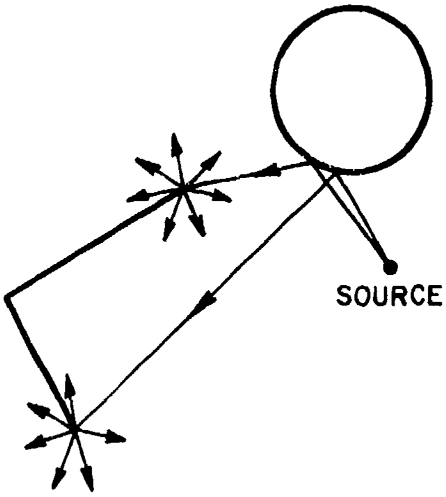


Figure 32--Illustration of ray reflected by cylinder and diffracted by plate edge.

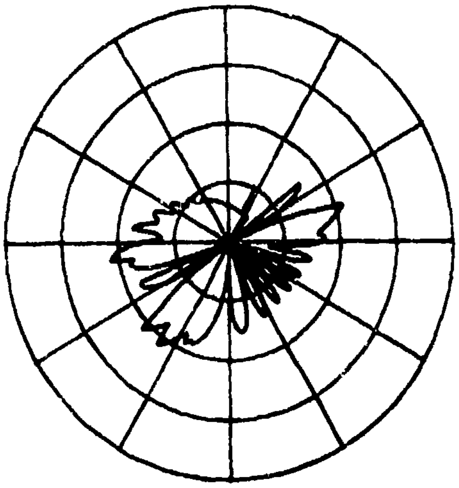


Figure 33--Fields reflected by cylinder, diffracted by plate edges.

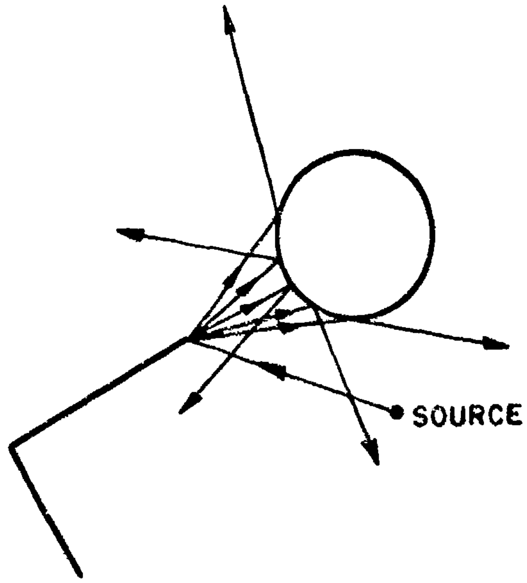


Figure 34--Illustration of rays diffracted by plate edge and reflected by cylinder.

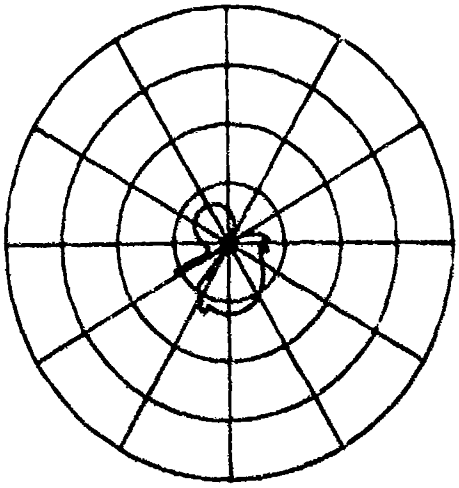


Figure 35--Fields diffracted by plate, reflected by cylinder.

The total pattern is obtained by summing the field components for the mechanisms mentioned previously. The total field pattern is illustrated in Figure 36.

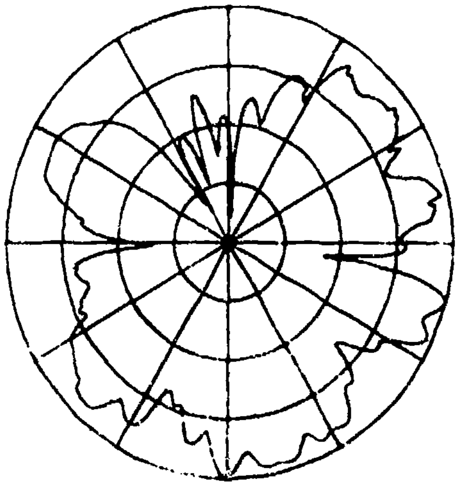


Figure 36--Total fields of source in the presence of scattering bodies.

Higher order scattering terms can also be computed, which will in some cases improve the accuracy of the field computations. Generally, it is found that such terms are negligible in magnitude as well as being difficult to compute and therefore are not included in the code. The presence of discontinuities in a final field pattern, however, indicates the presence of regions where higher-order terms are needed.

B. Method Used in Computing the Fields Using GTD

In order to use the Basic Scattering Code, the user first specifies a set of observation angles, for which he desires to obtain the far field pattern of the source(s) in the presence of a structure. The code computes the fields over the pattern angles specified for each source defined and uses superposition to obtain the total fields. For each observation direction computed, the code computes every GTD term applicable to the structure at hand, unless the user limits the types of terms computed. Each term is computed independently of the others. The following gives an outline of the procedure used in computing a particular GTD term.

The code first analyzes the input geometry in the geometry subroutines. Many of the parameters which do not vary for a given geometry are computed there in advance. This avoids re-computation of fixed variables. It also gives an a priori indication of the regions in which different GTD terms need to be included. This allows the code to avo'd performing computations where not necessary.

Two examples of GTD problems involving first and seconu order scattering phenomena are shown in Figures 37 and 38, respectively. A basic outline of how the various fields are computed is as follows:

1. Make any a priori checks of the fixed geometry
2. Compute ray path for specific mechanism desired
3. Determine if ray is blocked anywhere by another part of the structure
4. Use theory to compute the magnitude and phase of the field component resulting from the mechanism. If a second order mechanism is involved this is a two step process where the field of the first interaction is computed and then the field of the second interaction is computed.

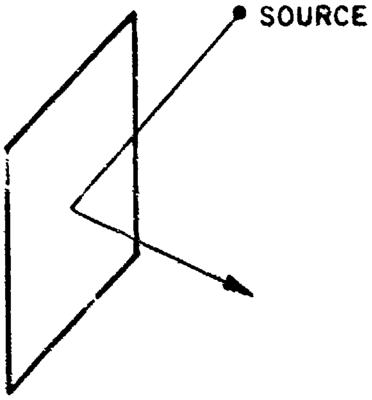


Figure 37--First order scattered term.

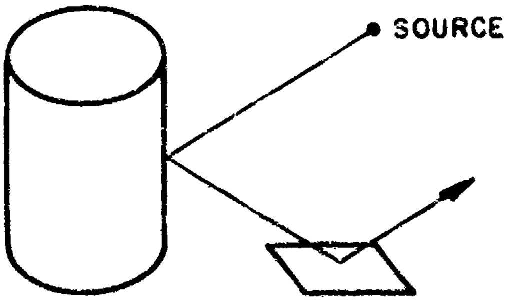


Figure 38--Second order scattered term.

The following is a more specific example of how the code computes the fields of a second order scattering term. The geometry, consisting of a source and four plates, is illustrated in Figure 39.

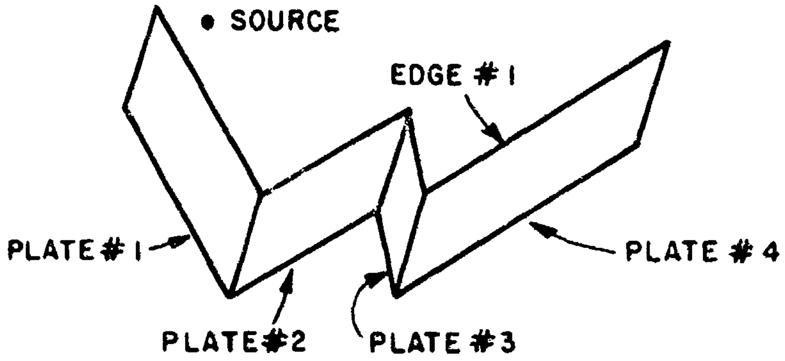


Figure 39--Illustration of a multiple plate example.

The code starts by choosing a source and a particular field term. As an example, let us choose the field reflected by plate #1 and then diffracted by edge #1 of plate #4. The code next chooses an observation angle and performs the following tasks:

1) The fixed geometry bounds are checked to see if a diffraction can occur in the direction specified. If it can, the code proceeds to the next step. If it can't, the code sets the fields to zero.

2) Determination of ray path. The code establishes the ray path, which includes both the reflection and diffraction points, as well as the propagation direction of the ray. It is temporarily assumed that plate #1 is of infinite extent, and that no shadowing occurs. It is also assumed that edge #1 of plate #4 is infinite. This guarantees that the reflection-diffraction will occur. The plate reflection can be handled by using the image of the source in plate #1 and removing the plate, as all the plate reflected rays appear to emanate from the image location (as shown in Figure 40).

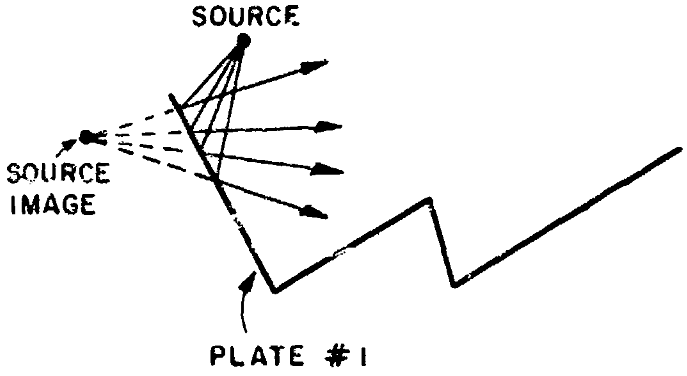


Figure 40--Illustration of the fields reflected off of plate #1.

The code then computes the diffraction point such that the law of diffraction is satisfied. The law of diffraction specifies that the angle the incident ray makes with the edge and the angle the diffracted ray makes with the edge are equal as shown in Figure 41.

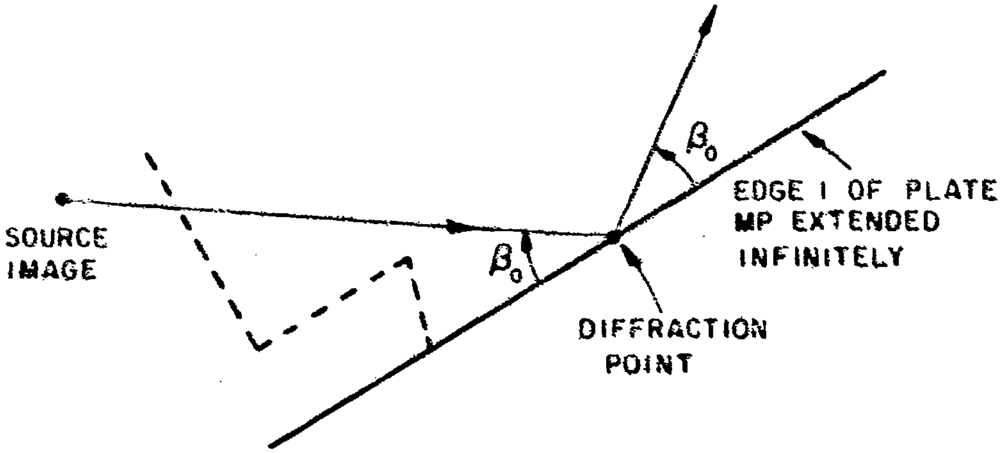


Figure 41--Illustration of a ray from the source image of plate #1 diffracted from edge #1 of plate #4.

Once the diffraction point is known, the reflection point on plate #1 is found by determining the line from the source image to the diffraction point. This reflection point is the intersection between this line and plate #1 as shown in Figure 42.

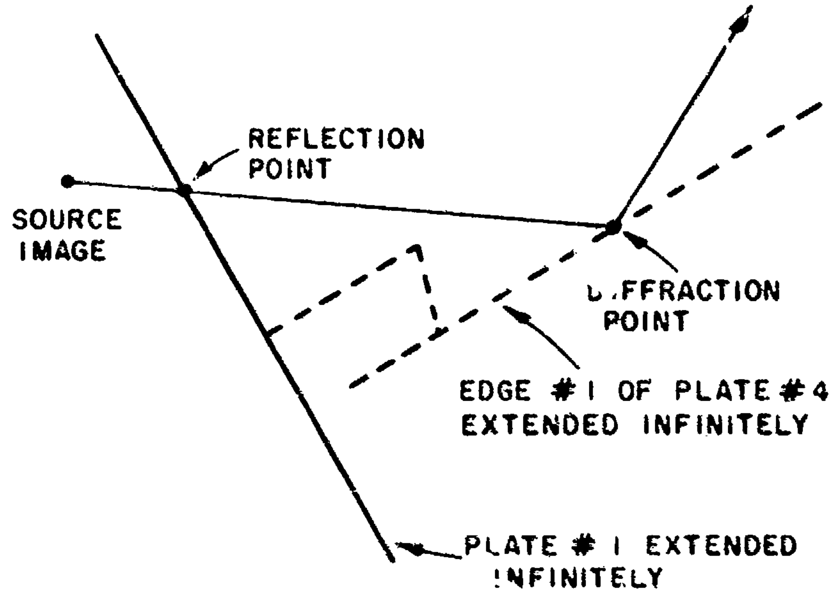
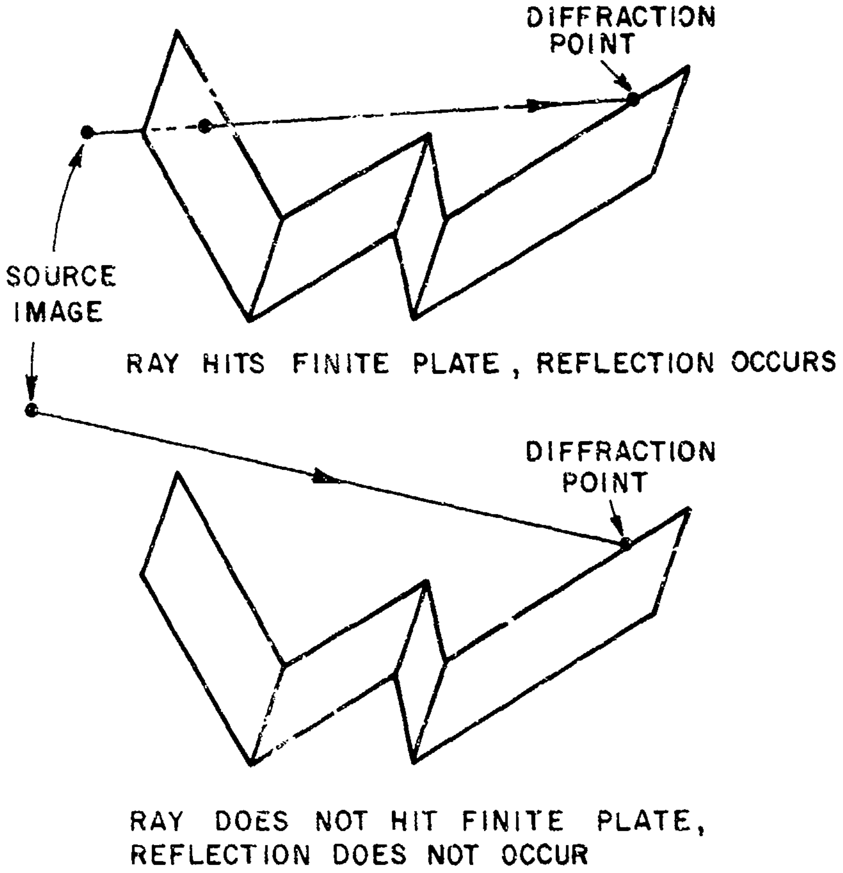

REFLECTION POINT
SOURCE IMAGE
DIFFRACTION POINT
EDGE # 1 OF PLATE # 4 EXTENDED INFINITELY
PLATE # 1 EXTENDED INFINITELY

Figure 42--Geometry used in finding reflection point on plate #1.

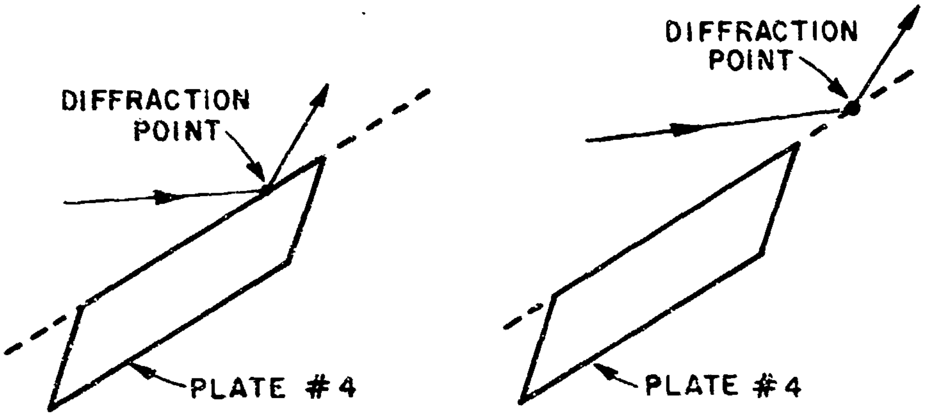
The code then determines if the ray path is valid for the finite geometry of the structure. The reflection point is valid if the line drawn from the image source to the diffraction point passes through the finite plate (plate #1) as shown in Figure 43. The diffraction point is valid if the point lies along the limits of the finite edge (see Figure 44). Figure 45 shows the computed ray path (assuming the scatterer points do lie on the finite plates).



RAY HITS FINITE PLATE, REFLECTION OCCURS

RAY DOES NOT HIT FINITE PLATE, REFLECTION DOES NOT OCCUR

Figure 43--Illustration of test for reflections from plate #1 for two different cases.



DIFFRACTION POINT ON EDGE, DIFFRACTION OCCURS

DIFFRACTION POINT NOT ON EDGE, DIFFRACTION DOES NOT OCCUR

Figure 44--Illustration of test for diffraction from edge #1 of plate #4.

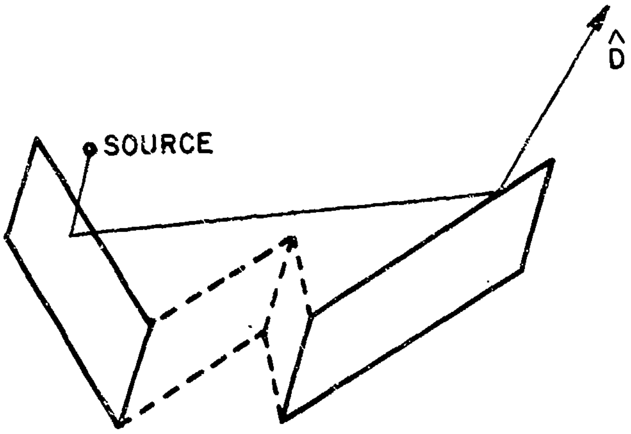
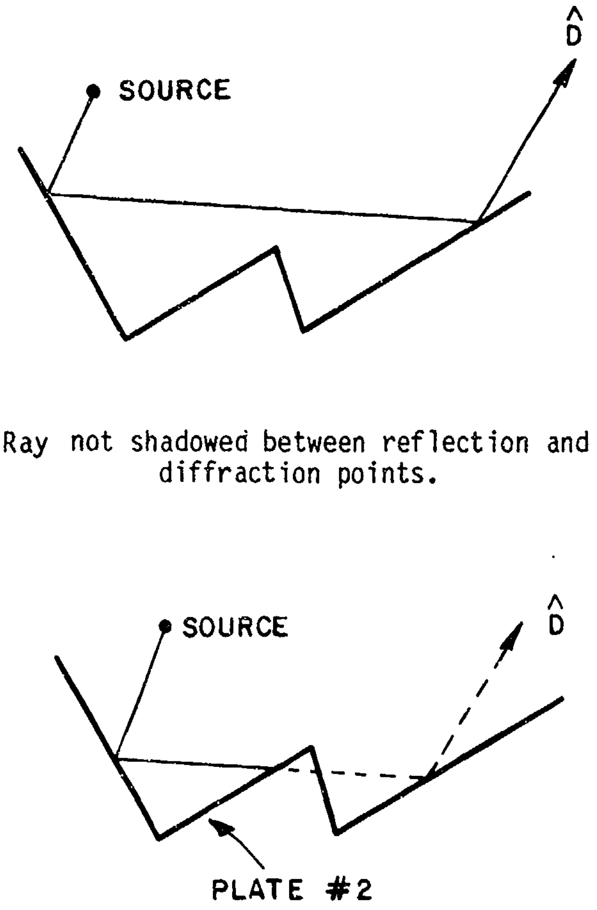


Figure 45--Illustration of the ray path for a field reflected from plate #1 then diffracted from edge #1 of plate #4.

3. Test for ray shadowing

The code next checks to see if the ray is obstructed anywhere along its path. The code first checks to see if the ray is shadowed after the diffraction by determining if a ray traveling in direction $\hat{D}$ from the diffraction point will hit a plate or a cylinder. The code then checks to see if the ray is shadowed between the reflection and diffraction points. In the example given this is the critical area, as there are other plates in this vicinity (see Figure 46).



Ray not shadowed between reflection and diffraction points.

Ray shadowed by plate #2.

Figure 46--Illustration of test for the shadowing of the ray of interest by a plate.

The code then checks to see if the source ray is shadowed by a plate or cylinder. If it is found that the ray is shadowed or that the reflection diffraction specified cannot happen for the geometry at hand, the code suspends computation of the term and sets the reflected-diffracted ray's field to zero.

4a. Computation of reflected field incident on the diffraction point.

The reflected field is simply obtained by using the image source located at the image position and computing the "free-space" fields incident on the diffraction point.

4b. Computation of the diffracted field

The diffracted field is obtained by multiplying the field incident on the edge by the edge diffraction coefficients. The parameters needed for the diffraction coefficients are obtained from the geometry of the incident and diffracted rays and the edge. The phase of the diffracted ray is then referred to the coordinate system origin and the task is completed.

# CHAPTER III
# CODE ORGANIZATION

The information in Chapter II is designed to present a qualitative view as to how the GTD is systematically used to construct a solution to a couple of problems. This chapter is intended to present how specific pieces of the code relate to the computation of the scattered fields.

First a brief outline is given in Section A. In Section B, tables are given, showing the interrelationships of the subroutines and the common blocks. Ways of reorganizing the code into smaller pieces are discussed in Section C. In Section D, a description of which variables must be redimensioned to change the maximum number of sources, plates, and edges is given.

In the last section, a brief discussion of most of the coordinate systems used in the code is presented. This is intended to provide a convenient way of reducing repetitive discussion of these systems throughout the subroutine descriptions.

## A. Overview of Code

The Basic Scattering Code is organized in a systematic way to increase the efficiency of computation by reducing core swapping and to allow different pieces to be run separately. This feature can be quite useful when it is necessary to run the code in a limited amount of core.

The various operations of the code are carried out in different classes of subroutines such as field computation, geometry, shadowing, ray tracing, and other service subroutines. Many of the subroutines are classified along with a brief description of their principal functions in Table 1.

The MAIN program provides the overall control of the various operations of the code. It controls the input of the geometry data, which is described in detail in the User's Manual[8]. It prepares the data for computation by transforming the input data into the optimum coordinate system for computations and by normalizing variables into common units of wavelengths. It also directs the computation of the various GTD fields and superimposes these fields to obtain the total far field pattern. The overall structure of the computation section of the code is outlined below.

The main computation section is composed of a number of large DO loops that step through all the various sources, GTD field types, scattering centers, and observation directions. Each loop ends at a common point where the different fields are superimposed and stored.

The first loop steps through the different sources. For a particular source the fixed properties of the geometry of the problem are first determined and stored. This includes the a priori bounds on the diffracted fields. These parameters are calculated in the geometry subroutines GEOM, GEOMC, and GEOMPC.

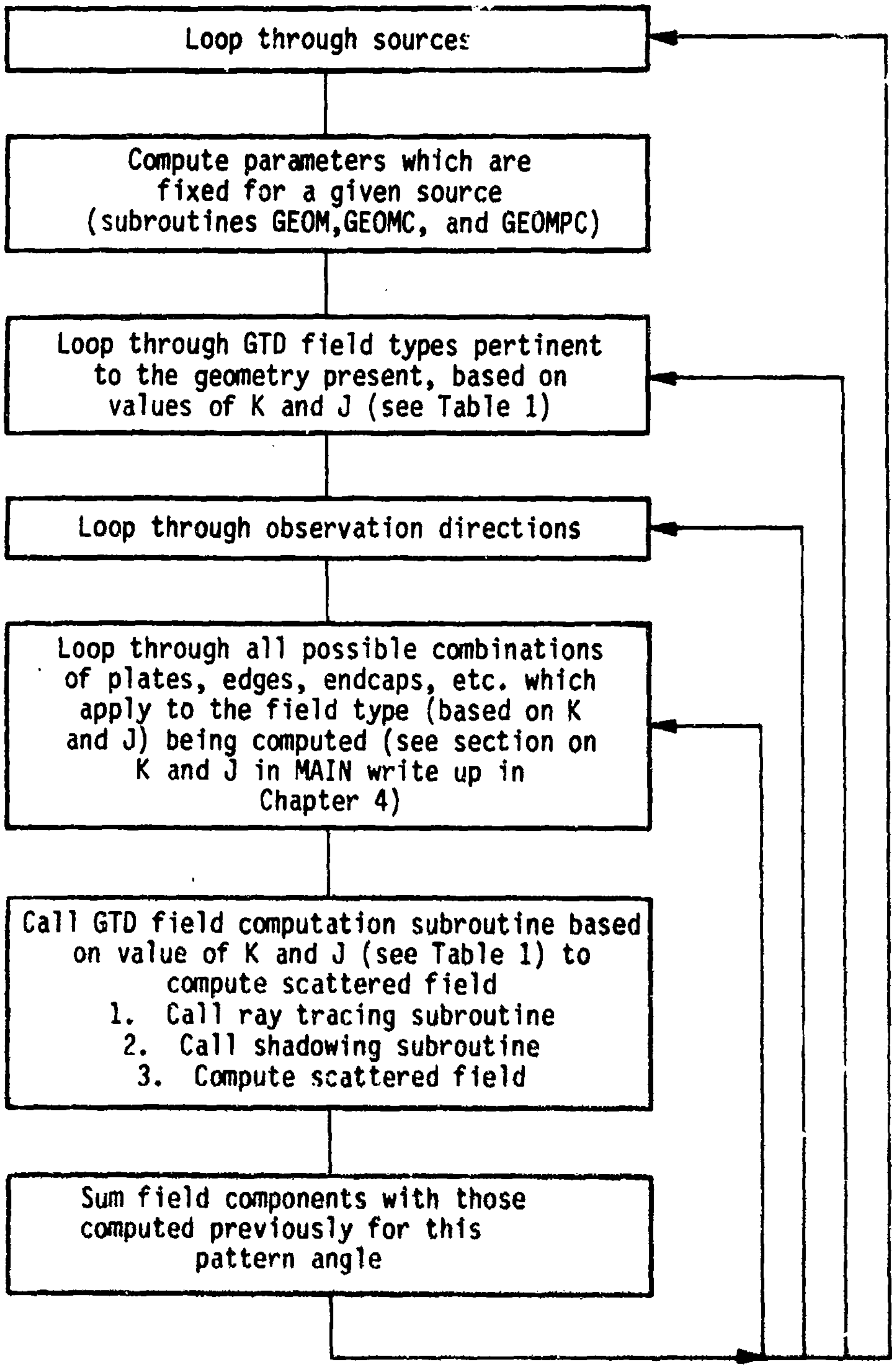
The MAIN program then loops through the various types of GTD fields which are identified by integers K and J as shown in Table 1. K=1 corresponds to fields involving plates, K=2 corresponds to cylinder fields, and K=3 to plate cylinder interaction fields. If only plates are present only the K=1 subroutines are called, if only a cylinder is present, only K=2 subroutines are called. If both plates and cylinders are present in the geometry, all three groups of subroutines (K=1,2,3) are called.

The MAIN program then loops through the various pattern angles desired. The observation directions are defined in the pattern cut coordinate system, discussed in Section III-E-4. Subroutine PATROT converts the observation direction to the reference coordinate system (RCS) discussed in Section III-E-1.

The MAIN program now branches to the section of the code where the specified GTD field subroutine is to be called. In this area the code loops through every combination of plates, edges, endcaps, etc. which apply to the GTD field (based on K and J) being computed. For example, if the fields reflected by one plate and then diffracted by another (K=1, J=5) are being computed, this section specifies every plate-edge combination in the geometry. This operation varies with the specific term being computed. Details are given in the MAIN program write up on K-J sections in Chapter IV-A. For each combination of plates, edges, etc., the MAIN program calls the appropriate field computation subroutine (listed in Table 1 according to K and J) to compute the scattered field.

This above arrangement of DO-loops is illustrated in the following flow diagram.

FLOW DIAGRAM


Loop through sources
Compute parameters which are fixed for a given source (subroutines GEOM,GEOMC, and GEOMPC)
Loop through GTD field types pertinent to the geometry present, based on values of K and J (see Table 1)
Loop through observation directions
Loop through all possible combinations of plates, edges, endcaps, etc. which apply to the field type (based on K and J) being computed (see section on K and J in MAIN write up in Chapter 4)
Call GTD field computation subroutine based on value of K and J (see Table 1) to compute scattered field
1. Call ray tracing subroutine
2. Call shadowing subroutine
3. Compute scattered field
Sum field components with those computed previously for this pattern angle

## TABLE 1
## LIST OF SOME IMPORTANT SUBROUTINES AND THEIR FUNCTION

| | | |
|---|---|---|
| Plate Field Subroutines K=1 | | |
| J=1 | INCFLD | - direct field |
| J=2 | REFPLA | - field reflected from a plate |
| J=3 | RPLRPL | - field doubly reflected by plates |
| J=4 | DIFPLT | - field diffracted by a plate |
| J=5 | RPLDPL | - field reflected by a plate then diffracted by a plate |
| J=6 | DPLRPL | - field diffracted by a plate then reflected by a plate |
| Cylinder Field Subroutines K=2 | | |
| J=1 | SCTCYL | - field scattered by a cylinder |
| J=2 | REFCAP | - field reflected by an end cap |
| J=3 | ENDIF | - field diffracted by an end cap rim |
| Plate-Cylinder Interaction Field Subroutines K=3 | | |
| J=1 | RPLSCL | - field reflected by a plate then scattered by a cylinder |
| J=2 | SCLRPL | - field scattered by a cylinder then reflected by a plate |
| J=3 | RCLDPL | - field reflected by a cylinder then diffracted by a plate |
| J=4 | DPLRCL | - field diffracted by a plate then reflected by a cylinder. |
| Geometry Subroutines | | |
| | GEOM | - fixed geometry of the plates |
| | GEOMC | - fixed geometry of the cylinder |
| | GEOMPC | - fixed geometry of the plate-cylinder interactions. |
| Shadowing Subroutines | | |
| | PLAINT | - shadowing due to plates |
| | CYLINT | - shadowing due to the cylinder |
| | CAPINT | - shadowing due to the end caps |
| Ray tracing Subroutine | | |
| | DFPTCL | - diffraction points on end cap rims |
| | DPTNFW | - near field diffraction point on plate edge |
| | DFPTWD | - far field diffraction point on plate edge |
| | DFRFPT | - diffraction point on plate edge then reflection point on cylinder |
| | RFDFPT | - reflection point on cylinder then diffraction point on plate edge |
| | RFPTCL | - far field reflection point on cylinder |
| | RFDFIN | - near field reflection point on cylinder |

The computation loops are nested in this manner, because for a given source and GTD field, a minimal number of subroutines need to be present in the computer core and they can stay there for a longer time than for any other loop configuration.

The field subroutines, whether they deal with reflection or diffraction for a plate or a cylinder, all have the same basic construction. The field subroutines start by checking the a priori bounds for the GTD mechanism of interest to see if it can produce a field propagating in the given observation direction. If it can, the code proceeds to trace the path back from the observation direction to the scattering points to the source without regard to the other structures in the geometry. This is accomplished in the ray tracing subroutines listed in Table 1. After the ray path is found, the path is tested to see if it is shadowed by any of the structures in the geometry. The shadowing subroutines are listed in Table 1. If the ray path has passed all the above tests the actual field calculation begins. First the field incident on the scattering point(s) is computed. The polarization is then converted to the proper canonical coordinate system for the particular GTD field. These coordinate systems are briefly discussed in section III-E. The use of the canonical system greatly simplifies the computation of the fields. Next, the reflection and/or the diffraction coefficients are computed for the problem. For example, the diffraction coefficient of the edge is computed in subroutine DW and its associated subroutines. The incident field is then multiplied by the reflection and/or diffraction coefficients along with the spread factors to compute the GTD field. The field polarization is then converted back to the reference coordinate system and the far field phase factor is added. This phase factor is the usual one given by

$$\frac{e^{-jks}}{s} \simeq e^{jk\overline{X}\cdot\hat{D}} \frac{e^{-jkR}}{R}$$

where s is the radial distance out from the scattering point, $\overline{X}$ is the location of the scattering point, $\hat{D}$ is the radial vector pointing in the observation direction, and R is the radial distance out from the reference coordinate system origin in the far field observation direction. This is illustrated in Figure 47. The factor $e^{-jkR}/R$ is suppressed at this point of the computations. It is added later in subroutine OUTPUT, for display purposes only, if the user specifies the distance R. The source weight, W, is also not added at this point in the calculations for convenience. The field therefore has the form

$$\overline{E} = W(E_\theta\hat{\theta}+E_\phi\hat{\phi}) \frac{e^{-jkR}}{R} ,$$

where $E_\theta$ and $E_\phi$ are calculated in the field computation subroutines.

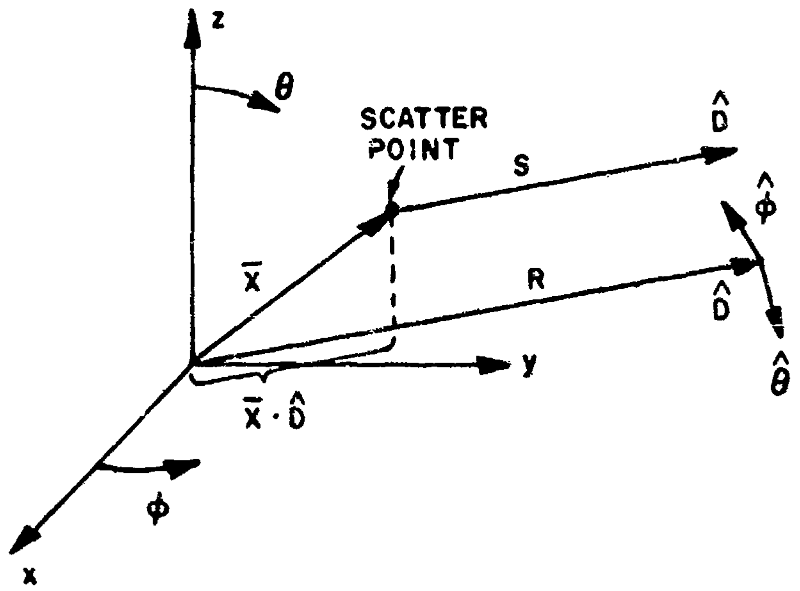


Figure 47. Illustration of the phase factor and polarization direction relative to the reference coordinate system.

The components $E_\theta$ and $E_\phi$ are returned to the MAIN program where they are superimposed on the fields from other scattering paths of the same mechanism. The individual fields can be printed out, if an in-depth analysis of a problem is desired. This is accomplished by setting the logical variable LOUT true in the input set. The subroutine PRIOUT displays the field appropriately identified by a code number (see the write up on subroutine PRIOUT).

The code then superimposes the fields for all the GTD terms as they are computed, weighted by their appropriate source weights and stores them by observation angle in two arrays based on polarization. When all the source, GTD fields, and pattern cut loops are finished the total result is printed out in various representations by the subroutine OUTPUT. The fields can also be plotted in various forms at this point. The code then loops back to the input section to accept a variation in the present geometry or a whole new problem.

## B. Subroutine and Common Block Linkage

In addition to the subroutines that have been classified in Table 1, there are many important subroutines that provide necessary services to other subroutines and to the code in general. The linkage of these is quite complicated because of the interdependence of many of the subroutines. A list of all the subroutines and the subroutines in which they are used is given in Table 2.

The majority of the data information transferred in the code is done by named common blocks. This method provides the most efficient and direct scheme for the large amounts of information present through the intertwined subroutines of the code. A list of all the COMMON BLOCKS and the subroutines which use them is given in Table 3.

# TABLE 2
# LINKAGE BETWEEN SUBROUTINES

| SUBROUTINE | SUBROUTINES IN WHICH IT IS USED |
|---|---|
| MAIN | Not applicable |
| BABS | MAIN,DFPTCL,DI,OUTPUT,POLYRT,PRIOUT |
| BLOCK DATA | Not applicable |
| BLOG10 | OUTPUT |
| BTAN2 | MAIN,CAPINT,CYLINT,DIFPLT,DPLRCL,DPLRPL,ENDIF, GEOM,GEOMPC,OUTPUT,PATROT,PLAINT,PRIOUT,RCLDPL, RCLRPL,REFBP,REFCYL,RFDFIN,RFDFPT,RFPTCL,RPLDPL, RPLRCL,RPLSCL,SCLRPL,SCTCYL,SOURCE,TANG |
| CAPINT | CYLINT,GEOMC,REFCAP |
| CYLINT | DIFPLT,DPLRCL,DPLRPL,GEOM,INCFLD,RCLDPL,RCLRPL, REFPLA,RPLDPL,RPLRPL,SCLRPL |
| DFPTCL | ENDIF |
| DFPTWD | DIFPLT,DPLRPL,RPLDPL |
| DFRFPT | DPLRCL |
| DI | DIFPLT,DPLRPL,DW,RPLDPL |
| DIFPLT | MAIN |
| DPI | DW |
| DPLRCL | MAIN |
| DPLRPL | MAIN |
| DPTNFW | GEOMPC |
| DQG32 | FKARG,RPLSCL,SCLRPL,SCTCYL |
| DW | DIFPLT,DPLRCL,DPLRPL,RCLDPL,RPLDPL |
| DZ | ENDIF |
| ENDIF | MAIN |
| FCT | RPLSCL,SCLRPL,SCTCYL |
| FFCT | DI,DIFPLT,DPLRPL,RPLDPL |
| FKARG | RPLSCL,SCLRPL,SCTCYL |
| FKY | DZ |
| FRNELS | DI,DPI,FFCT,FKY,RPLSCL,SCLRPL,SCTCYL |
| FUNI | FKARG |
| GEOM | MAIN |
| GEOMC | MAIN |
| GEOMPC | MAIN |
| IMAGE | DPLRPL,GEOM,RCLRPL,SCLRPL |
| IMCDIR | GEOMC |
| IMDIR | GEOM,RPLRPL |
| INCFLD | MAIN |
| NANDB | DPLRCL,ENDIF,RCLDPL,RCLRPL,REFCYL,RPLRCL,RPLSCL, SCLRPL,SCTCYL |
| OUTPUT | MAIN |
| PATROT | MAIN |
| PFUN | RPLSCL,SCLRPL,SCTCYL |
| PLAINT | DIFPLT,DPLRCL,DPLRPL,ENDIF,GEOM,INCFLD,RCLDPL, RCLRPL,REFCAP,REFCYL,REFPLA,RPLDPL,RPLRCL, RPLRPL,RPLSCL,SCLRPL,SCTCYL |
| POLYRT | DFPTCL,RFDFIN |
| PRIOUT | MAIN |
| QFUN | RPLSCL,SCLRPL,SCTCYL |
| RADCV | RPLSCL,SCLRPL,SCTCYL |
| RCLDPL | MAIN |
| RCLRPL | SCLRPL |
| REFBP | DPLRPL,RCLRPL,REFPLA,RPLDPL,RPLRCL,RPLRPL,RPLSCL, SCLRPL |
| REFCAP | MAIN |
| REFCYL | SCTCYL |
| REFPLA | MAIN |
| RFDFIN | GEOMPC |
| RFDFPT | RCLDPL |
| RFPTCL | RCLRPL,REFCYL,RPLRCL |
| ROTRAN | MAIN |
| RPLDPL | MAIN |
| RPLRCL | RPLSCL |
| RPLRPL | MAIN |
| RPLSCL | MAIN |
| SCLRPL | MAIN |
| SCTCYL | MAIN |
| SOURCE | DIFPLT,DPLRCL,DPLRPL,ENDIF,INCFLD,RCLDPL,RCLRPL, REFCAP,REFCYL,REFPLA,RPLDPL,RPLRCL,RPLRPL, RPLSCL,SCLRPL,SCTCYL |
| SOURCP | DIFPLT,DPLRPL,RPLDPL |
| TANG | CYLINT,GEOMC,GEOMPC |

# TABLE 3
# LINKAGE BETWEEN COMMON BLOCKS AND SUBROUTINES

| COMMON BLOCK | SUBROUTINES IN WHICH IT IS USED |
|---|---|
| BNDDCL | DFRFPT,DPLRCL,GEOMPC |
| BNDFCL | DIFPLT,DPLRCL,DPLRPL,GEOM,GEOMPC |
| BNDICL | GEOMPC,RFPTCL,RPLRCL,RPLSCL |
| BNDRCL | GEOMPC,RCLDPL,RFDFPT |
| BNDSCL | CYLINT,GEOMC,GEOMPC,RCLRPL,REFCYL,RFPTCL, SCLRPL,SCTCYL |
| BRNPHW | DFRFPT,GEOMPC,RFDFPT |
| CLDRC | MAIN,DPLRCL |
| CLRDC | MAIN,RCLDPL |
| CLRFC | MAIN,REFCYL |
| CLRFI | MAIN,RPLRCL |
| CLRFS | MAIN,RCLRPL |
| COMP | MAIN,BLOCK DATA,ENDIF,INCFLD,RPLSCL, SCLRPL,SCTCYL |
| DIP | MAIN,DFPTCL,DFRFPT,DIFPLT,DPLCRL,DPLRPL,ENDIF, INCFLD,RCLDPL,RCLRPL,REFCAP,REFCYL,REFPLA,RFDFPT, RPLDPL,RPLRCL,RPLRPL,RPLSCL,SCLRPL,SCTCYL |
| DOUBLE | MAIN,DIFPLT |
| EDMAG | DIFPLT,DPLRPL,GEOM,GEOMPC,RPLDPL |
| ESTOR | MAIN |
| FARP | MAIN,GEOM,GEOMC,GEOMPC,IMCDIR,IMDIR,SOURCE,SOURCP |
| FEDDAT | SOURCE |
| FNANG | MAIN,GEOM,GEOMPC |
| FUDG | REFCYL,SCTCYL |
| FUDGI | RPLRCL,RPLSCL |
| FUDGJ | RCLRPL,SCLRPL |
| GEOMEL | MAIN,CAPINT,CYLINT,DFPTCL,DFRFPT,DPLRCL,ENDIF, FKARG,FUNI,GEOMC,GEOMPC,NANDB,RADCV,RCLDPL, RCLRPL,REFCAP,REFCYL,RFDFIN,RFDFPT,RFPTCL, RPLRCL,RPLSCL,SCLRPL,SCTCYL,TANG |
| GEOPLA | MAIN,DFPTWO,DFRFPT,DIFPLT,DPLRCL,DPLRPL,DPTNEW, GEOM,GEOMPC,IMAGE,IMDIR,PLAINT,RCLDPL,RCLRPL, REFBP,REFPLA,RFDFPT,RFPTCL,RPLDPL,RPLRCL, RPLRPL,SCLRPL |
| GROUND | MAIN,GEOM,PLAINT,RFPTCL |
| GTD | MAIN,FCT,RADCV,RPLSCL,SCLRPL,SCTCYL |
| HITPLT | DIFPLT,GEOM,PLAINT |
| IMAINF | MAIN,GEOM,GEOMPC,REFPLA,RFPTCL,RPLDPL,RPLRCL, RPLRPL,RPLSCL |
| IMCINF | GEOMC,GEOMPC,REFCAP |
| LDCBY | MAIN,GEOMPC |
| LOGDIF | MAIN,DIFPLT,DPLRPL,RPLDPL |
| LPLCY | MAIN,CYLINT,GEOMC,GEOMPC,PLAINT |
| LSHDP | GEOM,GEOMAC,PLAINT |
| LSHDT | MAIN,GEOM,GEOMPC |
| OUTPTD | MAIN,OUTPUT |
| PATDAT | MAIN,PATROT |
| PIS | MAIN,BLOCK DATA,BTAN2,CYLINT,DFPTCL,DFRFPT, DI,DIFPLT,DPI,DPLRCL,DPLRPL,DZ,ENDIF,FCT,FFCT, FKARG,FKY,FRNELS,GEOM,GEOMC,GEOMPC,INCFLD, OUTPUT,PATROT,PFUN,PLAINT,PRIOUT,QFUN,RCLDPL, RCLRPL,REFBP,REFCAP,REFCYL,REFPLA,RFDFPT,RFPTCL, RPLDPL,RPLRCL,RPLRPL,RPLSCL,SCLRPL,SCTCYL,SOURCE, SOURCP,TANG |
| ROTRDT | MAIN,ROTRAN |
| SORINF | MAIN,DFPTCL,DFRFPT,DIFPLT,DPLRCL,DPLRPL,ENDIF, GEOM,GEOMC,GEOMPC,INCFLD,RCLDPL,RCLRPL,REFCAP, REFCYL,REFPLA,RFDFIN,RFDFPT,RFPTCL,RPLDPL, RPLRCL,RPLRPL,RPLSCL,SCLRPL,SCTCYL |
| SOURSF | MAIN,GEOM,GEOMC,GEOMPC,SOURCE,SOURCP |
| SRFACE | MAIN,GEOMC,GEOMPC |
| SURFAC | MAIN,DIFPLT,DPLRPL,GEOM,GEOMPC,RPLDPL |
| TEST | MAIN, CAPINT,CYLINT,DI,DIFPLT,DPI,DPLRCL,DPLRPL, ENDIF,FRNELS,GEOM,GEOMC,GEOMPC,PATROT,PLAINT, RCLDPL,RCLRPL,REFCAP,REFCYL,REFPLA,ROTRAN, RPLDPL,RPLRCL,RPLRPL,RPLSCL,SCLRPL,SCTCYL |
| THPHUV | MAIN, DIFPLT,DPLRCL,DPLRPL,ENDIF,INCFLD,RCLDPL, RCLRPL,REFCYL,RPLDPL,RPLRCL,SCLRPL |
| TOPD | BLOCK DATA,DI,DPI |

## C. Overlay Techniques

The Basic Scattering Code is a relatively large size computer code. Even though there are no big arrays, the large amount of coding requires a fairly significant amount of core in which to run. On an Amdahl computer, the code requires over 250 K bytes of core. The code has been designed, however, with overlay techniques in mind. Overlaying of the code can be accomplished by using the built in overlay techniques of a computer system, or by breaking the code up into smaller independent pieces that can be run separately with the results being superimposed later.

The code can be quite easily decomposed into three pieces. The three sections are composed of the subroutines necessary for plate fields, cylinder fields, and plate-cylinder interaction fields. The subroutines required to compute the plate fields are given in Table 4. If only plates are present in the geometry, the subroutines that are starred would not be necessary. The starred subroutines provide the shadowing algorithms for the cylinder. The subroutines required to compute the cylinder fields are given in Table 5. Similarly, the starred subroutines are necessary only if plate shadowing is desired. The subroutines required to compute the plate-cylinder interaction fields are given in Table 6. It is possible that for a particular problem or a particular computer system other techniques of separating the program would be more practical. The linkage information in Tables 2 and 3 should provide helpful information to accomplish such a task.

## TABLE 4
## SUBROUTINES USED IN PLATE COMPUTATIONS

BABS
BLOCK DATA
BLOG10
BTAN2
CAPINT*
CYLINT*
DFPTWD
DI
DIFPLT
DPI
DPLRPL
DW
FFCT
FRNELS
GEOM
GEOMC*
IMAGE
IMCDIR*
IMDIR
INCFLD
OUTPUT
PATROT
PLAINT
PRIOUT
REFBP
REFPLA
ROTRAN
RPLDPL
RPLRPL
SOURCE
SOURCP
TANG*

*Non-essential unless a cylinder is present in the geometry.

## TABLE 5
## SUBROUTINES USED IN CYLINDER COMPUTATIONS

BABS
BLOCK DATA
BLOG10
BTAN2
CAPINT
CYLINT
DFPTCL
DQG32
DZ
ENDIF
FCT
FKARG
FKY
FRNELS
FUNI
GEOM*
GEOMC
IMAGE*
IMCDIR
IMDIR*
INCFLD
NANDB
OUTPUT
PATROT
PFUN
PLAINT*
POLYRT
PRIOUT
QFUN
RADCV
REFCAP
REFCYL
RFPTCL
ROTRAN
SCTCYL
SOURCE
TANG

*Non-essential unless plates are present in the geometry.

## TABLE 6
## SUBROUTINES USED IN PLATE-CYLINDER INTERACTION COMPUTATIONS

BABS
BLOCK DATA
BLOG10
BTAN2
CAPINT
CYLINT
DFRFPT
DI
DPLRCL
DPTNFW
DQG32
DW
FCT
FKARG
FRNELS
FUNI
GEOM
GEOMC
GEOMPC
IMAGE
IMCDIR
IMDIR

NANDB
OUTPUT
PATROT
PFUN
PLAINT
POLYRT
PRIOUT
QFUN
RADCV
RCLDPL
RCLRPL
REFBP
RFDFIN
RFDFPT
RFPTCL
ROTRAN
RPLRCL
RPLSCL
SCLRPL
SOURCE
TANG

### D. Dimensions for Sources, Plates, and Edges

The maximum number of sources, plates, and edges that the Basic Scattering Code can accept is not limited by the theory. Any number of sources, plates, and edges can be used if the storage capacity of certain variables are set correctly in the DIMENSION statements and COMMON statements of the MAIN program and subroutines.

In order to change the maximum number of sources the code can accept, the dimension of the following variables in the MAIN program must be changed:

IMS($M_s$), XSS($M_s$,3), VXSS(3,3,$M_s$), WM($M_s$), WP($M_s$), HS($M_s$), HAWS($M_s$),

THOZ($M_s$), PHOZ($M_s$), THOX($M_s$), PHOX($M_s$),

where $M_s$ is equal to the maximum number of sources to be used. The variable MSDX should also be set equal to the integer $M_s$ in the text of the code.

In order to change the number of plates and edges the code can accept, the dimensions of the following variables must be changed in the indicated subroutines' DIMENSION statements and in the COMMON statements. The location of all the commons can be found in Table 3. In MAIN, the dimensions of the variable XX($M_p$,$M_e$,3) should be changed, where $M_p$ is equal to the maximum number of plates to be used and $M_e$ is the maximum number of edges allowed on each plate. The value of the variable MPDX should be set equal to $M_p$ and the variable MEDX should be set equal to $M_e$ in the text. In subroutine GEOM, the dimension of the variable IHIT($M_e$) should be changed. In subroutine RFPTCL the following dimensions should be changed:

IVD($M_{pp}$), PHOR($M_{pp}$), VRO($M_{pp}$), PHORP($M_{pp}$)

where $M_{pp} = 2M_p+1$.

In subroutine GEOMPC, the dimensions of the logical variable LCD($M_p$, $M_e$) should be changed. In the subroutines RFDFPT and DFRFPT the dimensions of the following variables should be changed:

IVD($M_p$,$M_e$), PHOR($M_p$,$M_e$), THOR($M_p$,$M_e$), VRO($M_p$,$M_e$), URO($M_p$,$M_e$),

PHORP($M_p$,$M_e$).

The variables in the following common blocks should be changed:

| | |
|---|---|
| BNDDCL: | VDC($M_p$,$M_e$), UDC(2), PDCR($M_p$,$M_e$,2)<br>TDCR($M_p$,$M_e$,2), DTDC($M_p$,$M_e$),<br>BTDC($M_p$,$M_e$,4), DDC($M_p$,$M_e$,2) |
| BNDFCL: | BD($M_p$,$M_e$,2) |
| BNDICL: | DTI($M_p$), VTI($M_p$,2), BTI($M_p$,4) |
| BNDRCL: | VCD($M_p$,$M_e$), UCD($M_p$,$M_e$), BCD($M_p$,$M_e$,2) |
| BRNPHW: | PHWR($M_p$,$M_e$) |
| CLDRC: | LDRC($M_p$,$M_e$) |
| CLRDC: | LRDC($M_p$,$M_e$) |
| CLRFI: | LRFI($M_p$) |
| CLRFS: | LRFS($M_p$) |
| DOUBLE: | IDD(361), ID($M_p$,$M_e$), II |
| EDMAG: | VMAG($M_p$,$M_e$) |
| FNANG: | FNP($M_p$,$M_e$) |
| GEOPLA: | X($M_p$,$M_e$,3), V($M_p$,$M_e$,3),<br>VP($m_p$,$M_e$,3), VN($M_p$,3),<br>MEP($M_p$),MPX |
| IMAINF: | XI($M_p$,$M_p$,3), VXI(3,3,$M_p$) |
| LDCBY: | LDC($M_p$,$M_e$) |
| LSHDP: | LSTS,LSTD($M_p$) |
| LSHDT: | LSHD($M_p$), LIHD($M_p$,$M_p$) |
| SURFAC: | LSURF($M_p$) |

## E. Coordinate Systems

In order to simplify the variety of geometrical calculations performed in the code, a number of different coordinate systems are used. Each system allows a certain set of computations to be performed with maximum ease. Each of these systems are defined in terms of the reference coordinate system (RCS).

### 1. Reference coordinate system

The reference coordinate system (RCS) is the fundamental system of the code. The system geometry is defined and stored in the RCS. Many of the calculations carried out are done in the RCS. It is therefore also referred to as the "computational coordinate system". Each of the other coordinate systems is defined in terms of the RCS (using RCS coordinates or unit vectors).

This coordinate system is the fixed system that the user defines the input geometry with respect to. However, if a cylinder is defined using the RT: command, a new reference coordinate system is established, before the computations begin. The z-axis of the new system coincides with the cylinder axis, the x-axis with the "A" dimension of the cylinder, and the y-axis with the "B" dimension of the cylinder. All the other input geometries are rotated and/or translated to this new RCS or "cylinder coordinate system" using subroutine ROTRAN. This is done to simplify the computation of the cylinder and plate-cylinder interaction fields. This transformation is not visible to the user in terms of the input or output parameters. The term "reference coordinate system" is, therefore, used for both the original system or the new system without distinguishing between them.

### 2. Definition coordinate system

Normally, the system geometry is defined by the user in the reference coordinate system. However, the user may choose to perform a coordinate system transformation (using the RT: command) in order to define part of the geometry in some preferred coordinate system. In using the RT: command, the user creates a "definition coordinate system" to which the data which follows the command will pertain. The user defines the definition coordinate system by specifying the origin location and direction of the $x_d$ and $z_d$ axes unit vectors. The origin of the definition system is defined at point $\overline{TR} = \hat{x}\ TR(1)+\hat{y}\ TR(2)+\hat{z}\ TR(3)$ in x,y,z RCS coordinates. The $\hat{x}_d$ unit vector of the definition system is defined by theta and phi angles THXP and PHXP as if it were a radial vector. The $\hat{z}_d$ unit vector is likewise defined by theta and phi angles THZP and PHZP in the RCS. The quantities TR, THXP, PHXP,THZP, and PHZP are all specified by the user. Note that all geometry defined in a definition coordinate system is immediately transformed into RCS notation using the method outlined in "transformation between systems" of this manual.

If the user defines the cylinder in a definition coordinate system (other than the RCS), the location of the system origin is stored along with the unit vectors of the definition system axes. The main program will later perform a transformation on the entire system geometry so that a new RCS is created where the z-axis of the new system coincides with cylinder axis. This new RCS is used for computational purposes.

### 3. Edge-fixed coordinate system

The code generates an edge-fixed coordinate system for each edge on every plate. The three (rectangular) coordinate system axes for each edge are positioned as follows:

1. in the plate plane and normal to the edge (the edge binormal)

2. normal to the plate

3. along the plate edge.

The unit vectors of the edge-fixed coordinate system axes are defined (using RCS unit vectors $\hat{x}$, $\hat{y}$, $\hat{z}$) as;

$$\widehat{VP} = \text{edge binormal} = \hat{x}\ VP(MP,ME,1) + \hat{y}\ VP(MP,ME,2) + \hat{z}\ VP(MP,ME,3)$$

$$\widehat{VN} = \text{plate normal} = \hat{x}\ VN(MP,1) + \hat{y}\ VN(MP,2) + \hat{z}\ VN(MP,3)$$

$$\hat{V} = \text{edge unit vector} = \hat{x}\ V(MP,ME,1) + \hat{y}\ V(MP,ME,2) + \hat{z}\ V(MP,ME,3).$$

Varables MP and ME specify which plate and which edge the unit vectors apply to.

The most significant use of the edge-fixed system is determining edge diffraction geometry. Incident and diffracted ray propagation angles along with polarization components are calculated by taking dot and cross products of edge-fixed unit vectors and ray propagation and polarization unit vectors. Edge-fixed unit vectors are also used in calculating geometry for intersecting plates, as well as checking to see if plates are flat.

The edge-fixed vectors are calculated in Section 2 of subroutine GEOM. Further details are given in this section.

### 4. Pattern cut coordinate system

The pattern cut coordinate system determines the axes about which the conical (theta fixed) or "orange-slice" (phi fixed) pattern cut is to be measured. Is is also the coordinate system in which the code output is given.

The user defines the pattern cut coordinate system by specifying the theta and phi angles which define the $x_p$ and $z_p$ axes of the system in the RCS (THCX,PHCX,THCZ,PHCZ, respectively). The user then specifies the type of cut to be made ($\theta_p$ or $\phi_p$ cut) and the range and increment of angles (in PD: section of MAIN).

The pattern cut system axes are stored in a 3x3 matrix of components which define the axes unit vectors in RCS components (see "Transformation between systems"). This matrix is used in the MAIN program in converting specific pattern angles from pattern cut coordinate system notation to the reference coordinate system (subroutine PATROT). Note that the pattern cut coordinate system is subject to the mass geometry transformation that is performed in the MAIN program if the user defines a cylinder in a coordinate system other than the RCS (using the RT: command). This transformation, however is not visible to the user. Note also that <u>definition</u> of the pattern cut coordinate system is done <u>independently</u> of any RT: commands. The user always defines it in the RCS.

5. <u>Reflection plate coordinate system</u>

The reflection plate coordinate system is used to handle reflection from plates when image theory is not used (in subroutines DPLRPL, RCLRPL, and SCLRPL). Only two of the three rectangular axes unit vectors are used:

$\hat{VN}$ = plate normal (calculated in subroutine GEOM)

$\hat{VT}$ = vector tangent to plate and normal to incident (and reflected) ray propagation direction.

The unit vectors are used to convert polarization to and from reflection plate coordinate system (parallel and normal to plate).

6. <u>Source coordinate system</u>

The source coordinate system is the system in which the source is defined. There is one such system for each source, although only one appears in the computations at a given time. Each time another source is used the source coordinate system is redefined.

For a one-dimensional source, the dipole lies along the $z_p$ axis. If an aperture source is used, the source lies in the $x_p$-$z_p$ plane, centered about the origin. In both cases the source current flows in the $\hat{z}_p$ direction.

Note that in this code the source coordinate system is designated with the subscript "p". The system may also be referred to as the "primed" or "antenna" coordinate system.

The source coordinate system is defined by the user in the input part of the main program. It is redefined later in the main program within the source loop. The origin about which the source is centered, is located at $\overline{XS}=\hat{x}$ XS(1) + $\hat{y}$ XS(2) + $\hat{z}$ XS(3) in RCS components. The unit vectors of the system axes are defined by a 3x3 matrix VXS(NI,NJ) of components. (See section on coordinate transformations). More specific definitions and illustrations are given in the section for subroutine SOURCE in the code manual.

7. Source image coordinate system

In many cases the code uses image theory in computing fields reflected from a plate. This involves computing an image source from which the reflected rays will appear to originate (see section on subroutine REFPLA). Assuming the source dimensions are known, the image source (or "source image") may be determined by computing the source image location (subroutine IMAGE) and the source image orientation (subroutine IMDIR). As the source location and orientation are specified using a source coordinate system, the "source image coordinate system" is used to define the image source. The location of the source image ($\overline{XIS}$ = $\hat{x}$ XIS(1) + $\hat{y}$ XIS(2) + $\hat{z}$ XIS(3)) is the origin of the source image coordinate system and the axes are defined by unit vectors in the same manner as the source coordinate system.

8. Transformation between systems

The majority of transformations performed in the code involve situations where it is necessary to transform a vector from one coordinate system to another. This involves rotation of coordinate systems without translation. Transformation of vector $\vec{V}$ in RCS components to $\vec{V}'$ in some other coordinate system is performed as follows:

$$\begin{bmatrix} V'_x \\ V'_y \\ V'_z \end{bmatrix} = \begin{bmatrix} X_{11} & X_{12} & X_{13} \\ X_{21} & X_{22} & X_{23} \\ X_{31} & X_{32} & X_{33} \end{bmatrix} \begin{bmatrix} V_x \\ V_y \\ V_z \end{bmatrix}$$

where $\vec{V} = \hat{x}\, V_x + \hat{y}\, V_y + \hat{z}\, V_z$ is the vector defined in the reference coordinate system and $\vec{V}' = \hat{x}\, V'_x + \hat{y}\, V'_y + \hat{z}\, V'_z$ is the vector defined in the second coordinate system. Inverse transformations are done by multiplying the vector $\vec{V}'$ by the transpose of the rotation matrix as follows:

$$\begin{bmatrix} V_x \\ V_y \\ V_z \end{bmatrix} = \begin{bmatrix} X_{11} & X_{21} & X_{31} \\ X_{12} & X_{22} & X_{32} \\ X_{13} & X_{23} & X_{33} \end{bmatrix} \begin{bmatrix} V'_x \\ V'_y \\ V'_z \end{bmatrix} .$$

The unit vectors of the second coordinate system are defined in the reference coordinate system as follows:

$$\hat{x}' = \hat{x}\, X_{11} + \hat{y}\, X_{12} + \hat{z}\, X_{13}$$
$$\hat{y}' = \hat{x}\, X_{21} + \hat{y}\, X_{22} + \hat{z}\, X_{23}$$
$$\hat{z}' = \hat{x}\, X_{31} + \hat{y}\, X_{32} + \hat{z}\, X_{33} .$$

The matrix used to transform (rotate) the vector is generally referred to as "x,y,z components defining the (name of system) coordinate system axes unit vectors in RCS components". The individual matrix elements are determined as follows:

$$X_{mn} = \hat{x}'_m \cdot \hat{x}_n \quad \text{(i.e., } X_{13} = \hat{x}' \cdot \hat{z}\text{, etc).}$$

When transforming a point, it is necessary to perform a translation as well as a rotation of coordinate systems (if the origins of the systems are different). To handle these situations, the code translates the point before rotating. Transformation of point $\bar{P} = \hat{x}\, P_x + \hat{y}\, P_y + \hat{z}\, P_z$ in the reference coordinate system to point $\bar{P}' = \hat{x}\, P'_x + \hat{y}\, P'_y + \hat{z}\, P'_z$ in another coordinate system is done as follows:

$$\begin{bmatrix} P'_x \\ P'_y \\ P'_z \end{bmatrix} = \begin{bmatrix} X_{11} & X_{12} & X_{13} \\ X_{21} & X_{22} & X_{23} \\ X_{31} & X_{32} & X_{33} \end{bmatrix} \begin{bmatrix} P_x - X_{ox} \\ P_y - X_{oy} \\ P_z - X_{oz} \end{bmatrix} ,$$

where $\bar{X}_o = \hat{x}\, X_{ox} + \hat{y}\, X_{oy} + \hat{z}\, X_{oz}$ is the location of the origin of the second coordinate system in the reference coordinate system and $X_{11}$, $X_{12}$, etc. are as defined previously. The reverse transformaiton is performed as follows:

$$\begin{bmatrix} P_x \\ P_y \\ P_z \end{bmatrix} = \begin{bmatrix} X_{ox} \\ X_{oy} \\ X_{oz} \end{bmatrix} + \begin{bmatrix} X_{11} & X_{21} & X_{31} \\ X_{12} & X_{22} & X_{32} \\ X_{13} & X_{23} & X_{33} \end{bmatrix} \begin{bmatrix} P'_x \\ P'_y \\ P'_z \end{bmatrix} .$$

# CHAPTER IV
# CODE DESCRIPTION

This section is divided into two sections, the first of which describes the operation of the MAIN program in detail. The second describes the subroutine and function operations. For each subroutine the following is given (as appropriate):

1. a statement of purpose
2. an illustration showing the geometry
3. a brief narrative of the method used
4. a flow diagram
5. a dictionary of major variables
6. a listing of the code.

The comment statements in the code listings follow the statements of the flow diagrams, simplifying correlation of the two.

## MAIN

PURPOSE

The main program reads the system geometry given by the user and directs the calculation of the scattered fields.

The main program is broken down into three parts as follows:

1. Data input section
2. Input conversion section
3. Main computation section.

Each of these sections is outlined on the following pages.

1. Data Input Section

The data input section reads an input file that contains the data specifying the geometry of the problem to be considered and prepares it for use by the code. The data input section is described in the User's Manual [8]. The commands available and the general flow of the input section is given in detail there, and will therefore not be repeated here.

```
C!!!
C!!!  THIS PROGRAM WAS WRITTEN AT THE OHIO STATE UNIVERSITY
C!!!  ELECTROSCIENCE LABORATORY. ANY PROBLEMS OR COMMENTS
C!!!  CAN BE REFERRED TO:
C!!!
C!!!       WALTER D. BURNSIDE (OR) RONALD J. MARHEFKA
C!!!       ELECTROSCIENCE LABORATORY
C!!!       1320 KINNEAR RD.
C!!!       COLUMBUS,OHIO 43212
C!!!     PHONE: (614) 422-5747 (OR) 422-5752
C!!!
C!!!  THIS PROGRAM COMPUTES THE FAR FIELD PATTERN OF AN ANTENNA
C!!!  OR SET OF ANTENNAS IN THE PRESENCE OF A SET OF PLATES
C!!!  AND/OR IN THE PRESENCE OF AN ELLIPTIC CYLINDER.
C!!!  THE PLATES ARE DEFINED BY THEIR CORNER LOCATIONS.
C!!!  AS DIMENSIONED, IT CAN HANDLE 14 PLATES WITH A MAXIMUM
C!!!  OF 6 CORNERS PER PLATE, AND 50 ANTENNA ELEMENTS CAN
C!!!  BE INPUT.  THE CYLINDER IS DEFINED BY ITS RADIUS
C!!!  ON ITS MAJOR AND MINOR AXIS AND THE END CAPS ARE
C!!!  DEFINED BY THEIR POSITION ON THE CYLINDER AXIS AND
C!!!  THE ANGLE OF THEIR SURFACES WITH THE CYLINDER AXIS IN
C!!!  THE X-Z CYL. PLANE.  NOTE THAT THE LIMITS ON THE NUMBER
C!!!  OF PLATES, CORNERS, AND SOURCES ARE ONLY DUE TO THE SIZE
C!!!  OF THE ARRAYS.  THE LINEAR DIMENSIONS ARE INPUT IN METERS
C!!!  UNLESS SPECIFIED.  THE ANGULAR DIMENSIONS ARE IN DEGREES.
C!!!
C!!!  NOTE THAT COMMENTS ARE GIVEN IN TWO FORMS:
C!!!       C!!! IMPLIES EXPLANATION OF PROGRAM SECTION
C!!!            IN TERMS OF THEORY
C!!!       C$$$ IMPLIES DESCRIPTION OF INPUT DATA
C!!!       C--- IMPLIES COMMAND INPUT READ SECTION
C!!!
C!!!  THIS VERSION WAS WRITTEN 8/2/79
C!!!
C!!!
      COMPLEX EITH,EIPH,ERTH,ERPH,ERPCT,ERPCP,ESTH,ESPH
      COMPLEX ERPST,ERPSP,ERPTH,ERPPH,ERRPT,ERRPP,ERCPT,ERCPP
      COMPLEX CJ,CPI4,WI,ETH,EPH,ETHT(361),EPHT(361)
      COMPLEX EDCTH,EDCPH,EDPTH,EDPPH,ERPDT,ERPDP,EDCRPT,EDCRPP
      COMPLEX EDRPT,EDRPP,ERDTH,ERDPH,EDRCT,EDRCP,ERCAT,ERCAP
      COMPLEX EDPCTH,EDPCPH,ERPDCT,ERPDCP,EDDTH,EDDPH,ERSPT,ERSPP
      DIMENSION IMS(50),XSS(50,3),VXSS(3,3,50),WV(50),WP(50)
      DIMENSION HS(50),HAWS(50),THOZ(50),PHOZ(50),THOX(50),PHOX(50)
      DIMENSION XX(14,6,3),XCO(3),XXCO(3),XOO(3),XXX(3)
      DIMENSION XPD(3),YPD(3),ZPD(3)
      DIMENSION JMN(3),JMX(3)
      DIMENSION LABEL(2,3),UNIT(3),IR(24),IT(14),ITT(10)
      DIMENSION XOR(3),TR(3),XP(3),YP(3),ZP(3),XO(3),XPP(3)
      LOGICAL LSOR,LOUT,LSRFC,LSURF,LSHD,LCYL,LPLA,LNROT
      LOGICAL LIHD,LDEBUG,LTEST,LSLOPE,LCORNR,LDC
      LOGICAL LWRITE,LPLT,LGRND,LAMP,LPRAD,LRANG,LCNPAT
      LOGICAL LRFC,LRFI,LRFS,LDRC,LRDC
      COMMON/DOUBLE/IDD(361),ID(14,6),II
      COMMON/FARP/IM,H,HAW
      COMMON/SOURSF/FACTOR
      COMMON/TEST/LDEBUG,LTEST
      COMMON/LOGDIF/LSLOPE,LCORNR
      COMMON/SORINF/XS(3),VXS(3,3)
      COMMON/IMAINF/XI(14,14,3),VXI(3,3,14)
      COMMON/PIS/PI,TPI,DPR,RPD
      COMMON/DIR/D(3),THSR,PHSR,SPS,CPS,STHS,CTHS
      COMMON/COMP/CJ,CPI4
      COMMON/THPHUV/DT(3),DP(2)
      COMMON/SURFAC/LSURF(14)
      COMMON/SRFACC/LSRFC(2)
      COMMON/LSHDT/LSHD(14),LIHD(14,14)
      COMMON/LDCBY/LDC(14,6)
      COMMON/FNANG/FNP(14,6)
      COMMON/GEOPLA/X(14,6,3),V(14,6,3),VP(14,6,3),VN(14,3)
     2,MFP(14),MPX
      COMMON/GEOMEL/A,B,ZC(2),SNC(2),CNC(2),CTC(2)
      COMMON/CTD/AS,IDC,SAS,SASP,CAS
      COMMON/ESTOR/ETHT,EPHT
      COMMON/LPLCY/LPLA,LCYL
      COMMON/GROUND/LGRND,MPXR
      COMMON/OUTPTD/LPRAD,LRANG,PRAD,RANG,WL
      COMMON/PATDAT/XPC(3),YPC(3),ZPC(3)
      COMMON/ROTRDT/XCL(3),YCL(3),ZCL(3)
      COMMON/CLRFC/LRFC
      COMMON/CLRFI/LRFI(14)
      COMMON/CLRFS/LRFS(14)
      COMMON/CLDRC/LDRC(14,6)
      COMMON/CLRDC/LRDC(14,6)
      DATA UNIT/1.,.3048,0.0254/
      DATA LABEL/'MET','ERS','FEE','T  ','INC','HES'/
      DATA IT/'TO:','PD:','PG:','SG:','LP:','PP:','GP:','XO:','RT:'
     2,'CG:','AM:','RG:','CM:','CE:'/
      DATA ITT/'UN:','FR:','NX:','EN:','NP:','NC:','NG:','NS:','PR:'
     2,'US:'/
C!!!  MAX. DIMENSION OF SOURCES,PLATES,AND EDGES.
      MSDX=50
      MPDX=14
      MEDX=6
C!!!  NOTE: IN SUB. RFPTCL THE VARIABLES IVD,PHOR,PHORP,AND VRO
C!!!          MUST BE DIMENSIONED 2*MPDX+1
      GO TO 2701
 2700 CONTINUE
      WRITE(6,3006)
      WRITE(6,3005)
 2701 CONTINUE
C!!!  INITIALIZE DATA TO DEFAULT VALUES.
      LDEBUG=.FALSE.
      LTEST=.FALSE.
      LOUT=.FALSE.
      LSLOPE=.TRUE.
      LCORNR=.TRUE.
      LSOR=.FALSE.
      LCYL=.FALSE.
      LPLA=.FALSE.
      LGRND=.FALSE.
      LWRITE=.TRUE.
      LPLT=.FALSE.
      LAMP=.FALSE.
      LPRAD=.FALSE.
      LRANG=.FALSE.
      RANG=1.
      PRAD=0.
      JMN(1)=1
      JMX(1)=7
      JMN(2)=1
      JMX(2)=3
      JMN(3)=1
      JMX(3)=4
      LCNPAT=.TRUE.
      TPPD=90.
      THCZ=0.
      PHCZ=0.
      THCX=90.
      PHCX=0.
      XPD(1)=1.
      XPD(2)=0.
      XPD(3)=0.
      YPD(1)=0.
      YPD(2)=1.
      YPD(3)=0.
      ZPD(1)=0.
      ZPD(2)=0.
      ZPD(3)=1.
      IB=0
      IE=300
      IS=1
      FRQG=.2997925
      MPX=0
      MEP(1)=4
      XX(1,1,1)=1.
      XX(1,1,2)=1.
      XX(1,1,3)=0.
      XX(1,2,1)=-1.
      XX(1,2,2)=1.
      XX(1,2,3)=0.
      XX(1,3,1)=-1.
      XX(1,3,2)=-1.
      XX(1,3,3)=0.
      XX(1,4,1)=1.
      XX(1,4,2)=-1.
      XX(1,4,3)=0.
      MSX=0
      XSS(1,1)=0.
      XSS(1,2)=0.
      XSS(1,3)=1.
      IMS(1)=0
      HS(1)=0.5
      HAWS(1)=0.
      THOZ(1)=0.
      PHOZ(1)=0.
      THOX(1)=90.
      PHOX(1)=0.
      VXSS(1,1,1)=1.
      VXSS(1,2,1)=0.
      VXSS(1,3,1)=0.
      VXSS(2,1,1)=0.
      VXSS(2,2,1)=1.
      VXSS(2,3,1)=0.
      VXSS(3,1,1)=0.
      VXSS(3,2,1)=0.
      VXSS(3,3,1)=1.
      WM(1)=1.
      WP(1)=0.
      RADIUS=3.
      IPLT=3
      THZP=0.
      PHZP=0.
      THXP=90.
      PHXP=0.
      TR(1)=0.
      TR(2)=0.
      TR(3)=0.
      XP(1)=1.
      XP(2)=0.
      XP(3)=0.
      YP(1)=0.
      YP(2)=1.
      YP(3)=0.
      ZP(1)=0.
      ZP(2)=0.
      ZP(3)=1.
      AA=1.
      BB=1.
      ZCN=-3.
      THTN=90.
      ZCP=3.
      THTP=90.
      XXCO(1)=0.
      XXCO(2)=0.
      XXCO(3)=0.
      XCL(1)=1.
      XCL(2)=0.
      XCL(3)=0.
      YCL(1)=0.
      YCL(2)=1.
      YCL(3)=0.
      ZCL(1)=0.
      ZCL(2)=0.
      ZCL(3)=1.
      IUNIT=1
      UNITS=UNIT(IUNIT)
      IUNST=0
      IUNSP=IUNST
      GO TO 2999
 3000 CONTINUE
      WRITE(6,3006)
 3006 FORMAT(1X,1H*,76X,1H*)
      WRITE(6,3006)
      WRITE(6,3005)
 3005 FORMAT(1X,26(3H***))
C!!!  READ IN VARIOUS COMMAND OPTIONS.
 2999 READ(5,3001,END=3004)(IR(I),I=1,24)
 3001 FORMAT(24A3)
      WRITE(6,3002)
 3002 FORMAT(1H ,//////,1X,26(3H***))
      WRITE(6,3006)
      WRITE(6,3003)(IR(I),I=1,24)
 3003 FORMAT(1X,1H*,2X,24A3,2X,1H*)
      IF(IR(1).EQ.IT(13))GO TO 3090
      IF(IR(1).EQ.IT(14))GO TO 3000
      WRITE(6,3006)
      WRITE(6,3006)
C!!!
C!!!  CHECK AGAINST STORED OPTIONS
C!!!
C!!!  CM: COMMENT CARD
C!!!  CE: LAST COMMENT CARD
C!!!  TO: TEST DATA GENERATION OPTION.
C!!!  UN: UNITS OF INPUT
C!!!  US: UNITS OF HS AND HAWS IN SG:
C!!!  FR: FREQUENCY
C!!!  PD: PATTERN DATA DESIRED
C!!!  PG: PLATE GEOMETRY INPUT
C!!!  SG: SOURCE GEOMETRY INPUT
C!!!  AM: NEC OR AMP INPUT
C!!!  PR: POWER RADIATED INPUT
C!!!  LP: LINE PRINTER LISTING OF RESULTS
C!!!  PP: PEN PLOT OF RESULTS
C!!!  GP: INCLUDE INFINITE GROUND PLANE
C!!!  XQ: EXECUTE PROGRAM
C!!!  RT: TRANSLATE AND/OR ROTATE COORDINATES
C!!!  CG: CYLINDER GEOMETRY INPUT
C!!!  RG: FAR FIELD RANGE INPUT
C!!!  NP: NEXT SET OF PLATES
C!!!  NG: NO GROUND PLANE
C!!!  NC: NO CYLINDER
C!!!  NS: NEXT SET OF SOURCES
```

```
C!!!  NX: NEXT PROBLEM
C!!!  EN: END PROGRAM
C!!!
      IF(IR(1).EQ.IT(1))GO TO 31ØØ
      IF(IR(1).EQ.IT(2))GO TO 32ØØ
      IF(IR(1).EQ.IT(12)) GO TO 325Ø
      IF(IR(1).EQ.IT(3))GO TO 33ØØ
      IF(IR(1).EQ.IT(4))GO TO 34ØØ
      IF(IR(1).EQ.IT(11))GO TO 345Ø
      IF(IR(1).EQ.IT(5))GO TO 35ØØ
      IF(IR(1).EQ.IT(6))GO TO 36ØØ
      IF(IR(1).EQ.IT(7))GO TO 37ØØ
      IF(IR(1).EQ.IT(8))GO TO 38ØØ
      IF(IR(1).EQ.IT(9))GO TO 39ØØ
      IF(IR(1).EQ.IT(1Ø))GO TO 4ØØØ
      IF(IR(1).EQ.ITT(1)) GO TO 41ØØ
      IF(IR(1).EQ.ITT(2)) GO TO 42ØØ
      IF(IR(1).EQ.ITT(3)) GO TO 27ØØ
      IF(IR(1).EQ.ITT(4)) GO TO 997
      IF(IR(1).EQ.ITT(5)) GO TO 335Ø
      IF(IR(1).EQ.ITT(6)) GO TO 4Ø5Ø
      IF(IR(1).EQ.ITT(7)) GO TO 375Ø
      IF(IR(1).EQ.ITT(8)) GO TO 349Ø
      IF(IR(1).EQ.ITT(9)) GO TO 344Ø
      IF(IR(1).EQ.ITT(1Ø)) GO TO 411Ø
      WRITE(6,3Ø21)
3Ø21  FORMAT(' ***   PROGRAM ABORTS!!! COMMAND INPUT IS NOT PART',
     1' OF STCRED COMMAND LIST   ***')
3ØØ4  STOP
C------
3Ø9Ø  CONTINUE
C------   CM:   CE:   COMMANDS   ------
C$$$
C$$$  IR(I)=CM: OR CE: FOLLOWED BY AN ALPHANUMERIC STRING OF
C$$$  CHARACTERS.  THE CM: COMMAND IMPLIES THAT THERE WILL BE
C$$$  ANOTHER COMMENT CARD FOLLOWING IT.  THE LAST COMMENT CARD
C$$$  MUST HAVE THE CE: COMMAND ON IT.  IF THERE IS ONLY ONE
C$$$  COMMENT CARD THE CE: COMMAND SHOULD BE USED.
C$$$
      READ(5,3ØØ1) (IR(I),I=1,24)
      WRITE(6,3ØØ3) (IR(I),I=1,24)
      IF(IR(1).EQ.IT(14)) GO TO 3ØØØ
      IF(IR(1).EQ.IT(13)) GO TO 3Ø9Ø
      WRITE(6,3Ø91)
3Ø91  FORMAT(' ***   PROGRAM ABORTS!!!   CE: COMMAND MUST BE',
     2' USED TO END COMMENTS.   ***')
      STOP
C------
31ØØ  CONTINUE
C-----   TO:   COMMAND   ------
C$$$
C$$$  LDEBUG=DEBUG DATA OUTPUT ON LINE PRINTER(TRUE OR FALSE)
C$$$
C$$$  LTEST=TEST DATA TO INSURE PROGRAM OPERATION(TRUE OR FALSE)
C$$$
C$$$  LOUT=OUTPUT MAIN PROGRAM DATA ON LINE PRINTER(TRUE OR FALSE)
C$$$
      READ(5,*) LDEBUG,LTEST,LOUT
      WRITE(6,31Ø1)LDEBUG,LTEST,LOUT
31Ø1  FORMAT(2H *,5X,'LDEBUG= ',L3,5X,'LTEST= ',L3,5X,'LOUT=',L3,
     1T79,1H*)
      WRITE(6,3ØØ6)
C$$$
C$$$  LSLOPE=SLOPE DIFFRACTED FIELD DESIRED (T OR F)
C$$$
C$$$  LCORNR=CORNER DIFFRACTED FIELD DESIRED (T OR F)
```

```
C$$$
C$$$  LSOR=ANTENNA PATTERN ALONE(TRUE OR FALSE)
C$$$
      READ(5,*)LSLOPE,LCORNR,LSOR
      WRITE(6,3102)LSLOPE,LCORNR,LSOR
 3102 FORMAT(2H *,5X,'LSLOPE= ',L3,5X,'LCORNR= ',L3,5X,'LSOR= ',L3,
     1T79,1H*)
      WRITE(6,3006)
      IF(LSOR)WRITE(6,3402)
 3402 FORMAT(2H *,5X,'SOURCE PATTERN ALONE IS COMPUTED!!!',T79,1H*)
      IF(LSOR)WRITE(6,3006)
C$$$
C$$$  JMN(1),JMX(1)=OPTION TO VARIOUS RAY TERMS FOR PLATES:
C$$$  0=SKIP PLATES SECTION
C$$$  1=INCIDENT FIELD
C$$$  2=SINGLE REFLECTED FIELD
C$$$  3=DOUBLE REFLECTED FIELD
C$$$  4=SINGLE DIFFRACTED FIELD
C$$$  5=REFLECTED/DIFFRACTED FIELD
C$$$  6=DIFFRACTED/REFLECTED FIELD
C$$$  NOTE: NORMALLY JMN(1)=1 AND JMX(1)=7. THIS COMPUTES ALL FIELD
C$$$  VALUES INCLUDING IDENTIFING DOUBLE DIFFRACTION PROBLEM AREAS
C$$$  FOR A CONVEX OR CONCAVE PLATE STRUCTURE.
C$$$
C$$$  JMN(2),JMX(2)=OPTION TO RUN VARIOUS RAY TERMS FOR CYLINDER:
C$$$  0=SKIP CYLINDER SECTION
C$$$  1=INCIDENT,REFLECTED,TRANSITION,AND CREEPING WAVE FIELDS
C$$$  2=SINGLE REFLECTED FIELDS FROM END CAPS
C$$$  3=SINGLE DIFFACTED FIELDS FROM END CAP RIMS
C$$$  NOTE: NORMALLY JMN(2)=1 AND JMX(2)=3. THIS COMPUTES ALL FIELD
C$$$  VALUES FOR A FINITE ELLIPTIC CYLINDER.
C$$$
C$$$  JMN(3),JMX(3)=OPTION TO RUN VARIOUS RAY TERMS FOR
C$$$  PLATE-CYLINDER INTERACTIONS:
C$$$  0=SKIP PLATE-CYLINDER INTERACTION SECTION
C$$$  1=FIELDS REFLECTED FROM THE PLATES THEN REFLECTED OR
C$$$  DIFFRACTED FROM THE CYLINDER
C$$$  2=FIELDS REFLECTED OR DIFFRACTED FROM THE CYLINDER THEN
C$$$  REFLECTED FROM THE PLATES
C$$$  3=FIELDS REFLECTED FROM THE CYLINDER THEN DIFFRACTED
C$$$  FROM THE PLATES
C$$$  4=FIELDS DIFFRACTED FROM THE PLATES THEN REFLECTED
C$$$  FROM THE CYLINDER
C$$$  NOTE: NORMALLY JMN(3)=1 AND JMX(3)=4.
C$$$
      READ(5,*) JMN(1),JMX(1),JMN(2),JMX(2),JMN(3),JMX(3)
      IF(JMN(1).LT.0) JMN(1)=1
      IF(JMX(1).GT.7) JMX(1)=7
      IF(JMN(2).LT.0) JMN(2)=1
      IF(JMX(2).GT.3) JMX(2)=3
      IF(JMN(3).LT.0) JMN(3)=1
      IF(JMX(3).GT.4) JMX(3)=4
      IF(LSOR) JMN(1)=1
      IF(LSOR) JMX(1)=1
      WRITE(6,3103) JMN(1),JMX(1),JMN(2),JMX(2),JMN(3),JMX(3)
 3103 FORMAT(2H *,2X,'JMN(1)= ',I2,2X,'JMX(1)= ',I2,2X,
     2'JMN(2)= ',I2,2X,'JMX(2)= ',I2,2X,'JMN(3)= ',I2,2X
     2,'JMX(3)= ',I2,T79,1H*)
      GO TO 3000
C------
 4100 CONTINUE
C---   UN:   COMMAND   ------
C$$$
C$$$   IUNIT=INDICATOR OF UNITS USED FOR INPUT DATA.
C$$$        1=METERS
C$$$        2=FEET
```

```
C$$$        3=INCHES
C$$$
      READ(5,*) IUNIT
      UNITS=UNIT(IUNIT)
      WRITE(6,41Ø1) (LABEL(J,IUNIT),J=1,2)
41Ø1  FORMAT(2H *,5X,'ALL THE LINEAR DIMENSIONS BELOW ARE'
     2,' ASSUMED TO BE IN ',2A3,T79,1H*)
      GO TO 3ØØØ
C-----
4110  CONTINUE
C---   US:   COMMAND   ------
C$$$
C$$$  IUNST=INDICATOR OF UNITS USED FOR HS AND HAWS IN THE
C$$$  SG: COMMAND.
C$$$        Ø=WAVELENGTHS
C$$$        1=METERS
C$$$        2=FEET
C$$$        3=INCHES
C$$$
C$$$  NOTE: IF ONE SOURCE IS SPECIFIED IN WAVELENGTHS, THEY ALL
C$$$           MUST BE IN WAVELENGTHS.
      READ(5,*) IUNST
      IF(MSX.EQ.Ø) GO TO 4112
      IF(IUNST.EQ.Ø.AND.IUNSP.EQ.Ø) GO TO 4112
      IF(IUNST.NE.Ø.AND.IUNSP.NE.Ø) GO TO 4112
      WRITE(6,4111)
4111  FORMAT(' *** PROGRAM ABORTS IN SOURCE UNITS. ALL UNITS NOT'
     2,' SPECIFIED IN WAVELENGTHS!!! ***')
      STOP
4112  CONTINUE
      IF(IUNST.EQ.Ø) GO TO 4114
      WRITE(6,4113) (LABEL(J,IUNST),J=1,2)
4113  FORMAT(2H *,5X,'THE SOURCE LENGTH HS AND WIDTH HAWS ARE'
     2,' ASSUMED TO BE IN ',2A3,T79,1H*)
      GO TO 4116
4114  WRITE(6,4115)
4115  FORMAT(2H *,5X,'THE SOURCE LENGTH HS AND WIDTH HAWS ARE'
     2,' ASSUMED TO BE IN WAVELENGTHS',T79,1H*)
4116  IUNSP=IUNST
      GO TO 3ØØØ
C-----
42ØØ  CONTINUE
C---   FR:   COMMAND   ------
C$$$
C$$$  FROG=FREQUENCY IN GIGAHERTZ.
C$$$
      READ(5,*) FROG
      WL=.2997925/FROG
      WRITE(6,42Ø1) FROG
42Ø1  FORMAT(2H *,5X,'FREQUENCY= ',F7.3,' GIGAHERTZ',T79,1H*)
      WRITE(6,3ØØ6)
      WRITE(6,42Ø2) WL
42Ø2  FORMAT(2H *,5X,'WAVELENGTH= ',F1Ø.6,' METERS',T79,1H*)
      GO TO 3ØØØ
C-----
32ØØ  CONTINUE
C---   PD:   COMMAND   ------
C$$$
C$$$  THCZ,PHCZ=ORIENTATION OF THE ZPD AXIS RELATIVE TO THE
C$$$  FIXED COORDINATE SYSTEM.
C$$$
C$$$  THCX,PHCX=ORIENTATION OF THE XPD AXIS RELATIVE TO THE
C$$$  FIXED COORDINATE SYSTEM
C$$$
      READ(5,*) THCZ,PHCZ,THCX,PHCX
      ZPD(1)=SIN(THCZ*RPD)*COS(PHCZ*RPD)
      ZPD(2)=SIN(THCZ*RPD)*SIN(PHCZ*RPD)
      ZPD(3)=COS(THCZ*RPD)
      XPD(1)=SIN(THCX*RPD)*COS(PHCX*RPD)
      XPD(2)=SIN(THCX*RPD)*SIN(PHCX*RPD)
      XPD(3)=COS(THCX*RPD)
C!!!  INSURE XPD IS PERPENDICULAR TO ZPD
      DZX=ZPD(1)*XPD(1)+ZPD(2)*XPD(2)+ZPD(3)*XPD(3)
      IF(ABS(DZX).GT.Ø.1) WRITE(6,32Ø1)
32Ø1  FORMAT(' *** PROGRAM ABORTS IN PATTERN CUT SECTION.'
     2,' THE COORDINATES ARE NOT ORTHOGONAL!!! ***')
      IF(ABS(DZX).GT.Ø.1) STOP
      XPD(1)=XPD(1)-ZPD(1)*DZX
      XPD(2)=XPD(2)-ZPD(2)*DZX
      XPD(3)=XPD(3)-ZPD(3)*DZX
      DOT=XPD(1)*XPD(1)+XPD(2)*XPD(2)+XPD(3)*XPD(3)
      DOT=SQRT(DOT)
      XPD(1)=XPD(1)/DOT
      XPD(2)=XPD(2)/DOT
      XPD(3)=XPD(3)/DOT
      YPD(1)=ZPD(2)*XPD(3)-ZPD(3)*XPD(2)
      YPD(2)=ZPD(3)*XPD(1)-ZPD(1)*XPD(3)
      YPD(3)=ZPD(1)*XPD(2)-ZPD(2)*XPD(1)
      WRITE(6,32Ø2)
32Ø2  FORMAT(2H *,5X,'THE PATTERN AXES ARE AS FOLLOWS:',T79,1H*)
      WRITE(6,3ØØ6)
      WRITE(6,32Ø3) (XPD(N),N=1,3)
32Ø3  FORMAT(2H *,5X,'XPD(1)=',F1Ø.5,'  XPD(2)=',F1Ø.5,'  XPD(3)='
     2,F1Ø.5,T79,1H*)
      WRITE(6,3ØØ6)
      WRITE(6,32Ø4) (YPD(N),N=1,3)
32Ø4  FORMAT(2H *,5X,'YPD(1)=',F1Ø.5,'  YPD(2)=',F1Ø.5,'  YPD(3)='
     2,F1Ø.5,T79,1H*)
      WRITE(6,3ØØ6)
      WRITE(6,32Ø5) (ZPD(N),N=1,3)
32Ø5  FORMAT(2H *,5X,'ZPD(1)=',F1Ø.5,'  ZPD(2)=',F1Ø.5,'  ZPD(3)='
     2,F1Ø.5,T79,1H*)
C$$$
C$$$  LCNPAT=IS PATTERN CONIC CUT(T OR F)?
C$$$  T=THETA CUT(CONIC CUT)
C$$$  F=PHI CUT(PHI CONSTANT)
C$$$
C$$$  TPPD=PATTERN ANGLE THAT IS CONSTANT
C$$$  IF LCNPAT=T: TPPD=THP CONSTANT
C$$$  IF LCNPAT=F: TPPD=PHP CONSTANT
C$$$
      READ(5,*) LCNPAT,TPPD
      WRITE(6,3ØØ6)
      IF(.NOT.LCNPAT) WRITE(6,32Ø6) TPPD
32Ø6  FORMAT(2H *,5X,'THETA IS BEING VARIED WITH PHI= ',F1Ø.5
     2,T79,1H*)
      IF(LCNPAT) WRITE(6,32Ø7) TPPD
32Ø7  FORMAT(2H *,5X,'PHI IS BEING VARIED WITH THETA= ',F1Ø.5
     2,T79,1H*)
      WRITE(6,3ØØ6)
C$$$
C$$$  IB,IE,IS=BEGIN,END,STEP
C$$$
      READ(5,*) IB,IE,IS
      IF(IB.LT.Ø) IB=Ø
      IF(IE.GT.36Ø) IE=36Ø
      IF(IS.LE.Ø) IS=1
      WRITE(6,32Ø8) IB,IE,IS
32Ø8  FORMAT(2H *,5X,'THE RANGE OF PATTERN ANGLE INDICES FOR THIS'
     2,' RUN ARE: ',I3,2(',',I3),T79,1H*)
      GO TO 3ØØØ
C------
```

```
 3250 CONTINUE
C---   RG:   COMMAND   ------
C$$$
C$$$  RANGS=FAR FIELD RANGE DISTANCE
C$$$
C$$$  NOTE IF RANGS IS GREATER THAN OR EQUAL TO 1.E3Ø
C$$$  THAN LRANG WILL BE SET FALSE
C$$$
      LRANG=.TRUE.
      READ(5,*) RANGS
      IF(RANGS.GT.9.9E29) GO TO 3252
      RANG=UNITS*RANGS
      WRITE(6,3251) RANGS,(LABEL(J,IUNIT),J=1,2),RANG
 3251 FORMAT(2H *,5X,'THE FAR FIELD RANGE SPECIFIED IS ',E12.6,
     2' IN ',2A3,T79,1H*,/2H *,5X,'THE RANGE SPECIFIED IN METERS'
     2,' IS ',E12.6,T79,1H*)
      GO TO 3ØØØ
 3252 CONTINUE
      LRANG=.FALSE.
      RANG=1.
      WRITE(6,3253)
 3253 FORMAT(2H *,5X,'NO FAR FIELD RANGE SPECIFIED.',T79,1H*)
      GO TO 3ØØØ
C------
 33ØØ CONTINUE
C---   PG:   COMMAND   ------
C$$$
C$$$  PLATE GEOMETRY INPUT
C$$$
      LPLA=.TRUE.
      MPX=MPX+1
      IF (MPX.GT.MPDX) WRITE(6,9Ø1) MPX
 9Ø1  FORMAT (' *****   NUMBER OF PLATES= ',I3,'  PROGRAM ABORTS',
     2' SINCE MAX. PLATE DIMENSION IS EXCEEDED.   *****')
      IF (MPX.GT.MPDX) STOP
      WRITE(6,33Ø1)MPX
 33Ø1 FORMAT(2H *,5X,'THIS IS PLATE NO. ',I3,' IN THIS ',
     1'SIMULATION.',T79,1H*)
      MP=MPX
      WRITE(6,3ØØ6)
      WRITE(6,3ØØ6)
      WRITE(6,3ØØ6)
C$$$
C$$$  MEP(MP)=NUMBER OF CORNERS ON THE MP-TH PLATE.
C$$$
      READ(5,*) MEP(MP)
      MEX=MEP(MP)
      IF (MEX.GT.MEDX) WRITE(6,9Ø3) MP,MEX
 9Ø3  FORMAT (' *****  PLATE #',I3,'  HAS ',I3,' EDGES.',
     2' PROGRAM ABORTS SINCE MAX. EDGE DIMENSION IS EXCEEDED.'
     2,' *****')
      IF(MEX.GT.MEDX) STOP
      DO 5 ME=1,MEX
C$$$
C$$$  XX(MP,ME,N)=X,Y,Z COMPONENTS OF CORNER #ME OF PLATE #MP.
C$$$  N=1(X),N=2(Y),N=3(Z). INPUT CORNER DATA AS FOLLOWS:
C$$$  1.,1.,Ø.
C$$$  -1.,1.,Ø.
C$$$  -1.,-1.,Ø.
C$$$  1.,-1.,Ø.
C$$$  THIS IS THE INPUT FOR A 2 METER SQUARE PLATE.
C$$$  NOTE THAT IF THERE IS MORE THAN ONE PLATE, THEN THE CORNER
C$$$  DATA FOR EACH PLATE WOULD FOLLOW SEQUENTIALLY.
C$$$
      READ(5,*) (XX(MP,ME,N),N=1,3)
      5    CONTINUE
           WRITE(6,33Ø2)(LABEL(J,IUNIT),J=1,2)
      33Ø2 FORMAT(2H *,2X,'PLATE#',2X,'CORNER#',3X,'INPUT LOCATION IN ',
          12A3,4X,'ACTUAL LOCATION IN METERS',T79,1H*)
           WRITE(6,33Ø3)
      33Ø3 FORMAT(2H *,2X,'------',2X,'-------'
          1,2(2X,2('-----------')),T79,1H*)
           DO 33Ø4 ME=1,MEX
           WRITE(6,3006)
           DO 331Ø N=1,3
      331Ø XO(N)=XX(MP,ME,N)
           DO 3311 N=1,3
      3311 XX(MP,ME,N)=UNITS*(XO(1)*XP(N)+XO(2)*YP(N)+XO(3)*ZP(N))+TR(N)
           WRITE(6,33Ø5)MP,ME,(XO(N),N=1,3),(XX(MP,ME,N),N=1,3)
      33Ø5 FORMAT(2H *,4X,I3,6X,I2,2X,2(2X,F8.3,2(',',F8.3)),T79,1H*)
      33Ø4 CONTINUE
           GO TO 3ØØØ
      C-----
      335Ø CONTINUE
      C--- NP:   COMMAND   -----
      C$$$
      C$$$ INITIALIZE PLATE DATA.
      C$$$
           LPLA=.FALSE.
           MPX=Ø
           WRITE(6,3351)
      3351 FORMAT(2H *,5X,' THE PLATE DATA IS INITIALIZED. ',T79,1H*/
          22H *,5X,' NO PLATES ARE PRESENTLY IN THE PROBLEM. ',T79,1H*)
           GO TO 3ØØØ
      C-----
      34ØØ CONTINUE
      C---  SG:   COMMAND   ------
      C$$$
      C$$$ MSX=NUMBER OF ANTENNA ELEMENTS.
      C$$$
           LAMP=.FALSE.
           MSX=MSX+1
           IF (MSX.GT.MSDX) WRITE(6,9Ø4) MSX
      9Ø4  FORMAT (' *****   NUMBER OF SOURCES= ',I3,'  PROGRAM',
          2' ABORTS SINCE MAX. SOURCE DIMENSION IS EXCEEDED.   *****')
           IF (MSX.GT.MSDX)STOP
           WRITE(6,34Ø1) MSX
      34Ø1 FORMAT(2H *,5X,'THISIS SOURCE NO. ',I3,' IN THIS',
          1' COMPUTATION.',T79,1H*)
           WRITE(6,3ØØ6)
           WRITE(6,3ØØ6)
      C$$$
      C$$$ XSS(MS,N)=XYZ LOCATION OF MS-TH ANTENNA ELEMENT.
      C$$$
      C$$$ IMS(MS)=TYPE OF LINEAR ANTENNA
      C$$$       Ø=ELECTRIC LINEAR ELEMENT
      C$$$       1=MAGNETIC LINEAR ELEMENT
      C$$$
      C$$$ HAMS(MS)=APERTURE WIDTH IN WAVELENGTHS   (NOTE: IF
      C$$$    HAMS(MS) IS LESS THAN .1 LAMBDA, SOURCE IS
      C$$$    CONSIDERED TO BE DIPOLE SOURCE
      C$$$ HS(MS)=LENGTH OF LINEAR ELEMENT IN WAVELENGTHS
      C$$$
      C$$$ THOZ(MS),PHOZ(MS)=ORIENTATION ANGLES USED TO DEFINE LINEAR
      C$$$ ELEMENT AXIS.
      C$$$
      C$$$ THOX(MS),PHOX(MS)=ORIENTATION ANGLES USED TO DEFINE APERTURE
      C$$$ PLANE OR DIPOLE X-AXIS.
      C$$$
      C$$$ AM(MS),AP(MS)=MAGNITUDE AND PHASE OF EXCITATION OF
      C$$$ MS-TH ELEMENT.
```

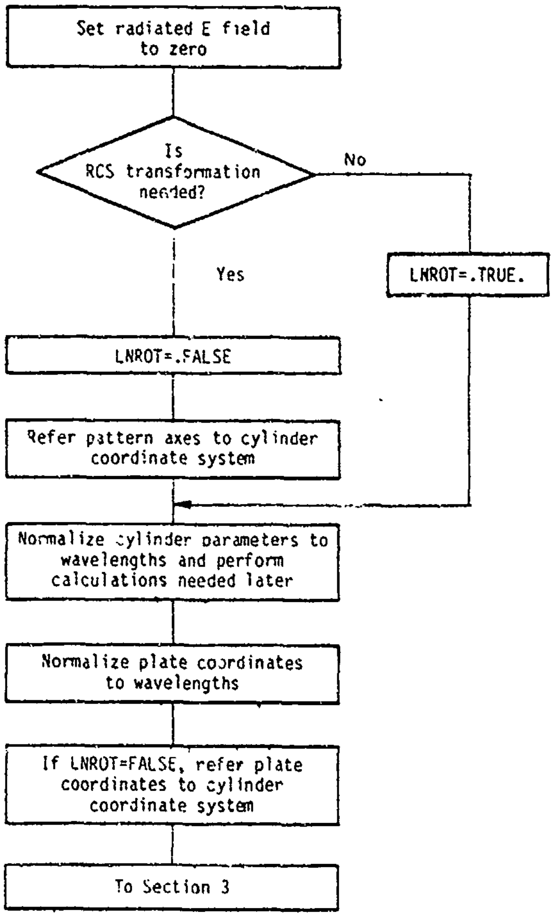
```
C$$$
      MS=MSX
      READ(5,*) (XSS(MS,N),N=1,3)
      READ(5,*) THOZ(MS),PHOZ(MS),THOX(MS),PHOX(MS)
      READ(5,*) IMS(MS),HS(MS),HAWS(MS)
      READ(5,*) WM(MS),WP(MS)
      IF(IMS(MS).EQ.Ø) WRITE(6,3411)
 3411 FORMAT(2H *,5X,'THIS IS AN ELECTRIC SOURCE.',T79,1H*)
      IF(IMS(MS).EQ.1) WRITE(6,3412)
 3412 FORMAT(2H *,5X,'THIS IS A MAGNETIC SOURCE.',T79,1H*)
      WRITE(6,3ØØ6)
      IF(IUNST.EQ.Ø) GO TO 3414
      UNSTS=UNIT(IUNST)
      WRITE(6,3413) HS(MS),HAWS(MS),(LABEL(J,IUNST),J=1,2)
 3413 FORMAT(2H *,5X,'SOURCE LENGTH=',F1Ø.5,' AND WIDTH='
     2,F1Ø.5,1X,2A3,T79,1H*)
      HS(MS)=UNSTS*HS(MS)
      HAWS(MS)=UNSTS*HAWS(MS)
      WRITE(6,3ØØ6)
      WRITE(6,3413) HS(MS),HAWS(MS),(LABEL(J,1),J=1,2)
      GO TO 3416
 3414 WRITE(6,3415) HS(MS),HAWS(MS)
 3415 FORMAT(2H *,5X,'SOURCE LENGTH=',F1Ø.5,' AND WIDTH='
     2,F1Ø.5,' WAVELENGTHS',T79,1H*)
 3416 WRITE(6,3ØØ6)
      WRITE(6,3417) WM(MS),WP(MS)
 3417 FORMAT(2H *,5X,'THE SOURCE WEIGHT HAS MAGNITUDE='
     2,F1Ø.5,' AND PHASE=',F1Ø.5,T79,1H*)
      WRITE(6,3ØØ6)
      WRITE(6,3ØØ6)
      WRITE(6,3421)(LABEL(J,IUNIT),J=1,2)
 3421 FORMAT(2H *,T6,'SOURCE#',T17,'INPUT LOCATION IN ',2A3,T46,
     1'ACTUAL LOCATION IN METERS',T79,1H*)
      WRITE(6,3422)
 3422 FORMAT(2H *,T6,7('-'),T16,27('-'),T45,27('-'),
     1T79,1H*)
      WRITE(6,3ØØ6)
      DO 3424 N=1,3
 3424 XO(N)=XSS(MS,N)
      DO 3425 N=1,3
 3425 XSS(MS,N)=UNITS*(XO(1)*XP(N)+XO(2)*YP(N)+XO(3)*ZP(N))+TR(N)
      WRITE(6,3426)MS,(XO(N),N=1,3),(XSS(MS,N),N=1,3)
 3426 FORMAT(2H *,T8,I3,T15,F8.3,2(',',F8.3),T44,F8.3,2(',',F8.3)
     1,T79,1H*)
      TOR=THOZ(MS)*RPD
      POR=PHOZ(MS)*RPD
      XO(1)=SIN(TOR)*COS(POR)
      XO(2)=SIN(TOR)*SIN(POR)
      XO(3)=COS(TOR)
      DO 3431 N=1,3
 3431 VXSS(3,N,MS)=XO(1)*XP(N)+XO(2)*YP(N)+XO(3)*ZP(N)
      TOR=THOX(MS)*RPD
      POR=PHOX(MS)*RPD
      XO(1)=SIN(TOR)*COS(POR)
      XO(2)=SIN(TOR)*SIN(POR)
      XO(3)=COS(TOR)
      DO 3432 N=1,3
 3432 VXSS(1,N,MS)=XO(1)*XP(N)+XO(2)*YP(N)+XO(3)*ZP(N)
      DZX=VXSS(1,1,MS)*VXSS(3,1,MS)+VXSS(1,2,MS)*VXSS(3,2,MS)
     2+VXSS(1,3,MS)*VXSS(3,3,MS)
      IF(ABS(DZX).GT.Ø.1) WRITE(6,3436)
 3436 FORMAT(' *** PROGRAM ABORTS IN SOURCE SECTION IN THAT THE',
     2' COORDINATES ARE NOT ORTHOGONAL !!! ***')
      IF(ABS(DZX).GT.Ø.1) STOP
      VXSS(1,1,MS)=VXSS(1,1,MS)-VXSS(3,1,MS)*DZX
      VXSS(1,2,MS)=VXSS(1,2,MS)-VXSS(3,2,MS)*DZX
      VXSS(1,3,MS)=VXSS(1,3,MS)-VXSS(3,3,MS)*DZX
      DOT=VXSS(1,1,MS)*VXSS(1,1,MS)+VXSS(1,2,MS)*VXSS(1,2,MS)
     2+VXSS(1,3,MS)*VXSS(1,3,MS)
      DOT=SQRT(DOT)
      VXSS(1,1,MS)=VXSS(1,1,MS)/DOT
      VXSS(1,2,MS)=VXSS(1,2,MS)/DOT
      VXSS(1,3,MS)=VXSS(1,3,MS)/DOT
      VXSS(2,1,MS)=VXSS(3,2,MS)*VXSS(1,3,MS)-VXSS(3,3,MS)*VXSS(1,2,MS)
      VXSS(2,2,MS)=VXSS(3,3,MS)*VXSS(1,1,MS)-VXSS(3,1,MS)*VXSS(1,3,MS)
      VXSS(2,3,MS)=VXSS(3,1,MS)*VXSS(1,2,MS)-VXSS(3,2,MS)*VXSS(1,1,MS)
      WRITE(6,3006)
      WRITE(6,3006)
      WRITE(6,3437)
 3437 FORMAT(2H *,5X,'THE FOLLOWING SOURCE ALIGNMENT IS USED:'
     2,T79,1H*)
      DO 3433 NI=1,3
      WRITE(6,3006)
 3433 WRITE(6,3434) (NI,NJ,MS,VXSS(NI,NJ,MS),NJ=1,3)
 3434 FORMAT(2H *,1X,3(2X,'VXSS(',I1,',',I1,',',I2,')=',F9.5)
     2,T79,1H*)
      GO TO 3000
C-----
 3440 CONTINUE
C---   PR:   COMMAND   -----
C$$$
C$$$  PRAD=TOTAL POWER RADIATED IN WATTS.
C$$$
C$$$  PRAD CAN ALSO BE SPECIFIED AS THE POWER INPUT IN WATTS.
C$$$
C$$$  NOTE IF PRAD IS LESS THAN OR EQUAL TO 1.E-30
C$$$  THAN LPRAD WILL BE SET FALSE
C$$$
      LPRAD=.TRUE.
      READ(5,*) PRAD
      IF(PRAD.LT.1.1E-30) GO TO 3442
      WRITE(6,3441) PRAD
 3441 FORMAT(2H *,5X,'TOTAL POWER RADIATED IN WATTS= ',E12.6
     2,T79,1H*)
      GO TO 3000
 3442 CONTINUE
      LPRAD=.FALSE.
      PRAD=0.
      WRITE(6,3443)
 3443 FORMAT(2H *,5X,'NO POWER RADIATED IS SPECIFIED',T79,1H*)
      GO TO 3000
C-----
 3450 CONTINUE
C---  AM:   COMMAND   ----
C$$$
C$$$  PRAD=TOTAL POWER RADIATED IN WATTS
C$$$
      LPRAD=.TRUE.
      READ(5,*) PRAD
      WRITE(6,3441) PRAD
      WRITE(6,3006)
C$$$
C$$$  MSX=NUMBER OF ANTENNA SEGMENTS
C$$$
      LAMP=.TRUE.
      READ(5,*) MSX
      IF(MSX.GT.MSDX) WRITE(6,3477) MSX
 3477 FORMAT(' *****   NUMBER OF SEGMENTS= ',I3,
     1' PROGRAM ABORTS SINCE MAX. SOURCE DIMENSION'
     2,' IS EXCEEDED.   *****')
      IF(MSX.GT.MSDX) STOP
      WRITE(6,3451) MSX
 3451  FORMAT(2H *,5X,'THERE ARE ',I3,' SEGMENTS IN THIS',
      2' COMPUTATION.',T79,1H*)
       WRITE(6,3006)
       WRITE(6,3006)
C$$$
C$$$  XS(MS,N)=XYZ LOCATION OF MS-TH ANTENNA SEGMENT
C$$$
C$$$  IMS(MS)=0=ELECTRIC LINEAR ELEMENT
C$$$
C$$$  HS(MS)=LENGTH OF LINEAR ELEMENT
C$$$
C$$$  THOZ(MS),PHOZ(MS)=ORIENTATION ANGLES USED TO DEFINE
C$$$  LINEAR ELEMENT AXIS.
C$$$
C$$$  WR(MS),WP(MS)=REAL AND IMAGINARY CURRENT WEIGHT.
C$$$
       WRITE(6,3006)
       WRITE(6,3454)
 3454  FORMAT(2H *,T31,'SEGMENT COORDINATES',T79,'*')
       WRITE(6,3006)
       WRITE(6,3006)
       WRITE(6,3456) (LABEL(J,IUNIT),J=1,2)
 3456  FORMAT(2H *,T7,'MS',T14,'INPUT LOCATION IN ',2A3,
      2T43,'ACTUAL LOCATION IN METERS',T79,'*')
       WRITE(6,3457)
 3457  FORMAT(2H *,T6,3('—'),T13,26('—'),T42,27('—'),T79,'*')
       WRITE(6,3006)
       DO 3452 MS=1,MSX
 3452  IMS(MS)=0
       DO 3119 MS=1,MSX
 3119  HAAS(MS)=0.
       DO 3453 MS=1,MSX
 3453  READ(5,*) (XSS(MS,N),N=1,3),HS(MS),THOZ(MS),PHOZ(MS)
       READ(5,*) (WM(MS),WP(MS),MS=1,MSX)
       DO 3473 MS=1,MSX
       DO 3474 N=1,3
 3474  XO(N)=XSS(MS,N)
       DO 3475 N=1,3
 3475  XSS(MS,N)=UNITS*(XO(1)*XP(N)+XO(2)*YP(N)+XO(3)*ZP(N))+TR(N)
       WRITE(6,3476) MS,(XO(N),N=1,3),(XSS(MS,N),N=1,3)
 3476  FORMAT(2H *,T6,I3,T13,F8.3,2(',',F8.3),T42,F8.3,
      22(',',F8.3),T79,1H*)
 3473  CONTINUE
       WRITE(6,3006)
       WRITE(6,3006)
       WRITE(6,3458) (LABEL(J,IUNIT),J=1,2)
 3458  FORMAT(2H *,T7,'MS',T13,'HS:',2A3,T23,'HS:METERS',
      2T41,'INPUT: THO,PHO',T60,'ACTUAL: THO,PHO',T79,1H*)
       WRITE(6,3459)
 3459  FORMAT(2H *,T6,3('—'),T12,20('—'),T40,16('—'),T59,
      217('—'),T79,1H*)
       WRITE(6,3006)
       DO 3463 MS=1,MSX
       HSO=HS(MS)
       HS(MS)=UNITS*HSO
       TO=90.-THOZ(MS)
       PO=PHOZ(MS)
       XO(1)=SIN(TO*RPD)*COS(PO*RPD)
       XO(2)=SIN(TO*RPD)*SIN(PO*RPD)
       XO(3)=COS(TO*RPD)
       DO 3481 N=1,3
 3481  XOR(N)=XO(1)*XP(N)+XO(2)*YP(N)+XO(3)*ZP(N)
       THOZ(MS)=DPR*ATAN2(SQRT(XOR(1)*XOR(1)+XOR(2)*XOR(2)),XOR(3))
       PHOZ(MS)=DPR*ATAN2(XOR(2),XOR(1))
       WRITE(6,3464) MS,HSO,HS(MS),TO,PO,THOZ(MS),PHOZ(MS)
 3464  FORMAT(2H *,T6,I3,3X,2(2X,F8.4),5X,2(2X,F8.3,',',F8.3)
     2,T79,1H*)
      DO 3484 N=1,3
3484  VXSS(3,N,MS)=XOR(N)
      VXSS(1,1,MS)=COS(THOZ(MS)*RPD)*COS(PHOZ(MS)*RPD)
      VXSS(1,2,MS)=COS(THOZ(MS)*RPD)*SIN(PHOZ(MS)*RPD)
      VXSS(1,3,MS)=-SIN(THOZ(MS)*RPD)
      VXSS(2,1,MS)=-SIN(PHOZ(MS)*RPD)
      VXSS(2,2,MS)=COS(PHOZ(MS)*RPD)
      VXSS(2,3,MS)=0.
3403  CONTINUE
      WRITE(6,3006)
      WRITE(6,3006)
      WRITE(6,3485)
3485  FORMAT(2H *,T33,'CURRENT WEIGHTS',T79,1H*,/2H *,T7,'MS',T18,
     2'REAL',T31,'IMAG.',T46,'MAG.',T57,'PHASE',T79,1H*)
      WRITE(6,3486)
3486  FORMAT(2H *,T6,3('-'),T17,6('-'),T30,7('-'),T45,6('-'),
     2T56,7('-'),T79,1H*)
      DO 3405 MS=1,MSX
      WMM=BABS(CMPLX(WR(MS),WP(MS)))
      WPP=DPR*BTAN2(WP(MS),WR(MS))
      WRITE(6,3466) MS,WR(MS),WP(MS),WPM,WPP
3466  FORMAT(2H *,T6,I3,5X,E11.4,2X,E11.4,4X,E11.4,2X,F8.3,T79,1H*)
3465  CONTINUE
      WRITE(6,3006)
      GO TO 3000
C------
3490  CONTINUE
C---  NS*   COMMAND   -----
C$$$
C$$$  INITIALIZE SOURCE DATA.
C$$$
      LAMP=.FALSE.
      MSX=0
      WRITE(6,3491)
3491  FORMAT(2H *,5X,' THE SOURCE DATA IS INITIALIZED. ',T79,1H*/
     22H *,5X,' NO SOURCES ARE PRESENTLY IN THE PROBLEM. '
     2,T79,1H*)
      GO TO 3000
C------
3500  CONTINUE
C---   LP*   COMMAND   -----
C$$$
C$$$  LWRITE=TRUE IF LINE PRINTER OUTPUT OF DATA IS DESIRED
C$$$
      READ(5,*) LWRITE
      IF(.NOT.LWRITE) WRITE(6,3505)
3505  FORMAT(2H *,5X,'NO LINE PRINTER OUTPUT',T79,1H*)
      IF(.NOT.LWRITE) GO TO 3000
      WRITE(6,3501)
3501  FORMAT(2H *,5X,' DATA WILL BE OUTPUT ON LINE PRINTER !!!',
     1T79,1H*)
      GO TO 3000
C------
3600  CONTINUE
C---   PP*   COMMAND   -----
C$$$
C$$$  LPLT=TRUE IF PEN PLOTTER OUTPUT IS DESIRED
C$$$
      READ(5,*) LPLT
      IF(.NOT.LPLT) WRITE(6,3606)
3606  FORMAT(2H *,5X,'NO PEN PLOT DESIRED',T79,1H*)
      IF(.NOT.LPLT) GO TO 3000
C$$$
C$$$  IF LPLT=TRUE READ IN DIMENSIONS
C$$$
```

```
C$$$  RADIUS=RADIUS OF POLAR PLOT.
C$$$  IPLT=1(FIELD PLOT), 2(POWER PLOT), 3(DB. PLOT)
C$$$
      READ(5,*) RADIUS,IPLT
      WRITE(6,3602)
3602  FORMAT(2H *,5X,'DATA WILL BE OUTPUT TO PEN PLOTTER !!!'
     2,T79,1H*)
      WRITE(6,3006)
      WRITE(6,3601)RADIUS,IPLT
3601  FORMAT(2H *,5X,' RADIUS=',F6.2,5X,'IPLT=',I3,T79,1H*)
      GO TO 3000
C------
3700  CONTINUE
C---   GP:   COMMAND   ------
C$$$
C$$$  INFINITE GROUND PLANE EFFECT INCLUDED.
C$$$
      LGRND=.TRUE.
      DO 3702 N=1,3
      XX(14,1,N)=1.E5*(XP(N)+YP(N))+TR(N)
      XX(14,2,N)=1.E5*(-XP(N)+YP(N))+TR(N)
      XX(14,3,N)=1.E5*(-XP(N)-YP(N))+TR(N)
3702  XX(14,4,N)=1.E5*(XP(N)-YP(N))+TR(N)
      WRITE(6,3701)
3701  FORMAT(2H *,5X,'INFINITE GROUND PLANE INSERTED IN',
     1' STRUCTURE !!!',T79,1H*)
      WRITE(6,3006)
      WRITE(6,3703) (TR(N),N=1,3)
3703  FORMAT(2H *,5X,'THE ORIGIN IS AT ',F12.6,',',F12.6
     2,',',F12.6,' METERS',T79,1H*)
      WRITE(6,3006)
      WRITE(6,3704) (ZP(N),N=1,3)
3704  FORMAT(2H *,5X,'THE NORMAL IS ',F12.6,',',F12.6,','
     2,F12.6,T79,1H*)
      GO TO 3000
C------
3750  CONTINUE
C---   NG:   ------
C$$$
C$$$  INITIALIZE GROUND PLANE DATA.
C$$$
      LGRND=.FALSE.
      WRITE(6,3751)
3751  FORMAT(2H *,5X,' GROUND PLANE DATA IS INITIALIZED. ',T79,1H*/
     22H *,5X,' NO GROUND PLANE IS PRESENTLY IN THE PROBLEM. '
     2,T79,1H*)
      GO TO 3000
C------
3900  CONTINUE
C---   MT:   COMMAND   ------
C$$$
C$$$  TR(N)=LINEAR TRANSLATION OF COORDINATES FROM THE FIXED
C$$$  COORDINATES WHICH IS ORIGINALLY SET UP BY OPERATOR.
C$$$
      READ(5,*) (TR(N),N=1,3)
3921  FORMAT(2H *,5X,' INPUT DATA GIVEN IN TERMS OF ',2A3,T79,1H*)
      WRITE(6,3901)((LABEL(J,IUNIT),J=1,2),(TR(N),N=1,3))
3901  FORMAT(2H *,5X,'TRANSLATION IN ',2A3,':  TR(1)=',F8.3,
     1'  TR(2)=',F8.3,'  TR(3)=',F8.3,T79,1H*)
      DO 3922 N=1,3
3922  TR(N)=TR(N)*UNITS
      WRITE(6,3006)
      IF(IUNIT.NE.1)WRITE(6,3901)((LABEL(J,1),J=1,2),(TR(N),N=1,3))
      IF(IUNIT.NE.1)WRITE(6,3006)
      WRITE(6,3006)
C$$$
```

```
C$$$  THZP,PHZP=ORIENTATION OF THE ZP AXIS RELATIVE TO THE
C$$$  FIXED COORDINATE SYSTEM.
C$$$
C$$$  THXP,PHXP=ORIENTATION OF THE XP AXIS RELATIVE TO THE
C$$$  FIXED COORDINATE SYSTEM.
C$$$
      READ(5,*)THZP,PHZP,THXP,PHXP
      ZP(1)=SIN(THZP*RPD)*COS(PHZP*RPD)
      ZP(2)=SIN(THZP*RPD)*SIN(PHZP*RPD)
      ZP(3)=COS(THZP*RPD)
      XP(1)=SIN(THXP*RPD)*COS(PHXP*RPD)
      XP(2)=SIN(THXP*RPD)*SIN(PHXP*RPD)
      XP(3)=COS(THXP*RPD)
C!!!  INSURE XP IS PERPENDICULAR TO ZP
      DZX=ZP(1)*XP(1)+ZP(2)*XP(2)+ZP(3)*XP(3)
      IF(ABS(DZX).GT.0.1)WRITE(6,3903)
 3903 FORMAT(' *** PROGRAM ABORTS IN ROTATE SECTION IN THAT THE',
     1' COORDINATES ARE NOT ORTHOGONAL!!! ***')
      IF(ABS(DZX).GT.0.1)STOP
      XP(1)=XP(1)-ZP(1)*DZX
      XP(2)=XP(2)-ZP(2)*DZX
      DOT=SQRT(DOT)
      XP(3)=XP(3)-ZP(3)*DZX
      DOT=XP(1)*XP(1)+XP(2)*XP(2)+XP(3)*XP(3)
      XP(1)=XP(1)/DOT
      XP(2)=XP(2)/DOT
      XP(3)=XP(3)/DOT
      YP(1)=ZP(2)*XP(3)-ZP(3)*XP(2)
      YP(2)=ZP(3)*XP(1)-ZP(1)*XP(3)
      YP(3)=ZP(1)*XP(2)-ZP(2)*XP(1)
      WRITE(6,3931)
 3931 FORMAT(2H *,5X,'THE FOLLOWING ROTATIONS ARE USED FOR ALL',
     1' SUBSEQUENT INPUTS:',T79,1H*)
      WRITE(6,3006)
      WRITE(6,3932)(XP(N),N=1,3)
 3932 FORMAT(2H *,5X,'XP(1)=',F10.5,'  YP(2)=',F10.5,'  XP(3)=',
     1F10.5,T79,1H*)
      WRITE(6,3006)
      WRITE(6,3933)(YP(N),N=1,3)
 3933 FORMAT(2H *,5X,'YP(1)=',F10.5,'  YP(2)=',F10.5,'  YP(3)=',
     1F10.5,T79,1H*)
      WRITE(6,3006)
      WRITE(6,3934)(ZP(N),N=1,3)
 3934 FORMAT(2H *,5X,'ZP(1)=',F10.5,'  ZP(2)=',F10.5,'  ZP(3)=',
     1F10.5,T79,1H*)
      GO TO 3000
C------
 4000 CONTINUE
C----   CYL   COMMAND   -----
C$$$
C$$$  CYLINDER GEOMETRY INPUT
C$$$
      LCYL=.TRUE.
C$$$
C$$$  AA=RADIUS OF ELLIPSE ON X CYLINDER AXIS
C$$$  BB=RADIUS OF ELLIPSE ON Y CYLINDER AXIS
C$$$
C$$$  ZCN,THCN=MOST NEGATIVE END CAP'S Z COMPONENT
C$$$  AND ANGLE OF SURFACE WITH THE NEG. CYLINDER AXIS
C$$$  ZCP,THCP=MOST POSITIVE END CAP'S Z COMPONENT
C$$$  AND ANGLE OF SURFACE WITH THE POS. CYLINDER AXIS
C$$$
      READ(5,*) AA,BB
      READ(5,*) ZCN,THCN,ZCP,THCP
      [illegible] AA
      [illegible]
      ZCNO=ZCN
      ZCPO=ZCP
      AA=AA*UNITS
      BB=BB*UNITS
      ZCN=ZCN*UNITS
      ZCP=ZCP*UNITS
      WRITE(6,6310)(LABEL(J,IUNIT),J=1,2),AAO
 6310 FORMAT(2H *,5X,'X AXIS DIMENSION IN ',
     22A3,'=',F8.3,T79,1H*)
      WRITE(6,3006)
      WRITE(6,6310) (LABEL(J,1),J=1,2),AA
      WRITE(6,3006)
      WRITE(6,3006)
      WRITE(6,6320)(LABEL(J,IUNIT),J=1,2),BBO
 6320 FORMAT(2H *,5X,'Y AXIS DIMENSION IN ',
     22A3,'=',F8.3,T79,1H*)
      WRITE(6,3006)
      WRITE(6,6320) (LABEL(J,1),J=1,2),BB
      WRITE(6,3006)
      WRITE(6,3006)
      WRITE(6,6330)(LABEL(J,IUNIT),J=1,2),ZCNO
 6330 FORMAT(2H *,5X,'MOST NEGATIVE END CAP Z COMPONENT IN ',
     22A3,'=',F8.3,T79,1H*)
      WRITE(6,3006)
      WRITE(6,6330) (LABEL(J,1),J=1,2),ZCN
      WRITE(6,3006)
      WRITE(6,6340) THTN
 6340 FORMAT(2H *,5X,'ANGLE OF NEG. END CAP SURFACE WITH NEG.',
     2' CYL. AXIS ','=',F8.3,T79,1H*)
      WRITE(6,3006)
      WRITE(6,3006)
      WRITE(6,6350)(LABEL(J,IUNIT),J=1,2),ZCPO
 6350 FORMAT(2H *,5X,'MOST POSITIVE END CAP Z COMPONENT IN ',
     22A3,'=',F8.3,T79,1H*)
      WRITE(6,3006)
      WRITE(6,6350) (LABEL(J,1),J=1,2),ZCP
      WRITE(6,3006)
      WRITE(6,6360) THTP
 6360 FORMAT(2H *,5X,'ANGLE OF POS. END CAP SURFACE WITH POS.',
     2' CYL. AXIS ','=',F8.3,T79,1H*)
      WRITE(6,3006)
      DO 6370 N=1,3
      XXCO(N)=TR(N)
      XCL(N)=XP(N)
      YCL(N)=YP(N)
 6370 ZCL(N)=ZP(N)
      GO TO 3000
C-----
 4050 CONTINUE
C---  NC*   COMMAND   -----
C$$$
C$$$  INITIALIZE CYLINDER DATA.
C$$$
      LCYL=.FALSE.
      WRITE(6,4051)
 4051 FORMAT(2H *,5X,' CYLINDER DATA IS INITIALIZED. ',T79,1H*/
     22H *,5X,' NO CYLINDER IS PRESENTLY IN THE PROBLEM. '
     2,T79,1H*)
      GO TO 1000
C-----
 597  CONTINUE
C---   END   COMMAND   -----
C$$$
C$$$  END PROGRAM
C$$$
      WRITE(6,3000)
      WRITE(6,3ØØ6)
      WRITE(6,3ØØ5)
      GO TO 999
C-----
 38ØØ CONTINUE
C--    XO:   COMMAND   -----
C$$$
C$$$  EXECUTE PROGRAM
C$$$
      WRITE(6,3ØØ6)
      WRITE(6,3ØØ6)
      WL=.2997925/FRQG
      WRITE(6,3ØØ5)
      MPXR=MPX
C!!!  GROUND PLANE IS ANOTHER PLATE IN SOLUTION.
      IF(LGRND)MPXR=MPX+1
      IF(MPXR.GT.MPDX)WRITE(6,9Ø1)MPXR
      IF(MPXR.GT.MPDX)GO TO 999
      IF(.NOT.LGRND)GO TO 38Ø1
      LPLA=.TRUE.
      MEP(MPXR)=4
      DO 38Ø2 I=1,4
      DO 38Ø2 N=1,3
 38Ø2 XX(MPXR,I,N)=WL*XX(MPDX,I,N)
 38Ø1 CONTINUE
      IF(MPXR.EQ.Ø) LPLA=.FALSE.
C!!!
```

## 2. Input Conversion Section

This section converts the input data into a preferred form for computational purposes. This involves converting angles in degrees to units of radians, linear dimensions into wavelengths, and performing the reference coordinate system (RCS) transformation if needed. The RCS transformation is done if a cylinder is present and its axis does not line up with the basic coordinate system used to define the input geometry.

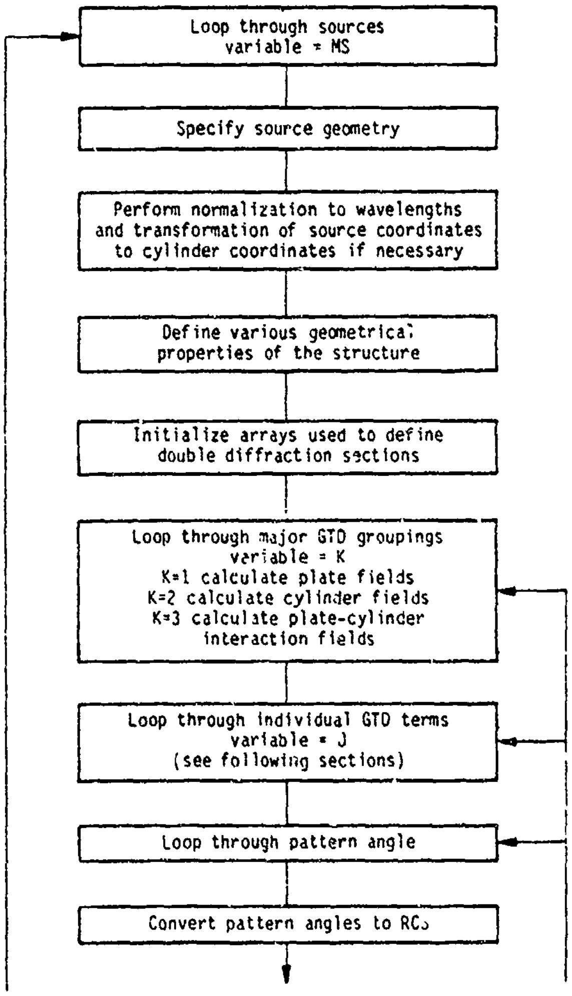

Set radiated E field to zero
Is RCS transformation needed?
No
Yes
LNROT=.TRUE.
LNROT=.FALSE
Refer pattern axes to cylinder coordinate system
Normalize cylinder parameters to wavelengths and perform calculations needed later
Normalize plate coordinates to wavelengths
If LNROT=FALSE, refer plate coordinates to cylinder coordinate system
To Section 3

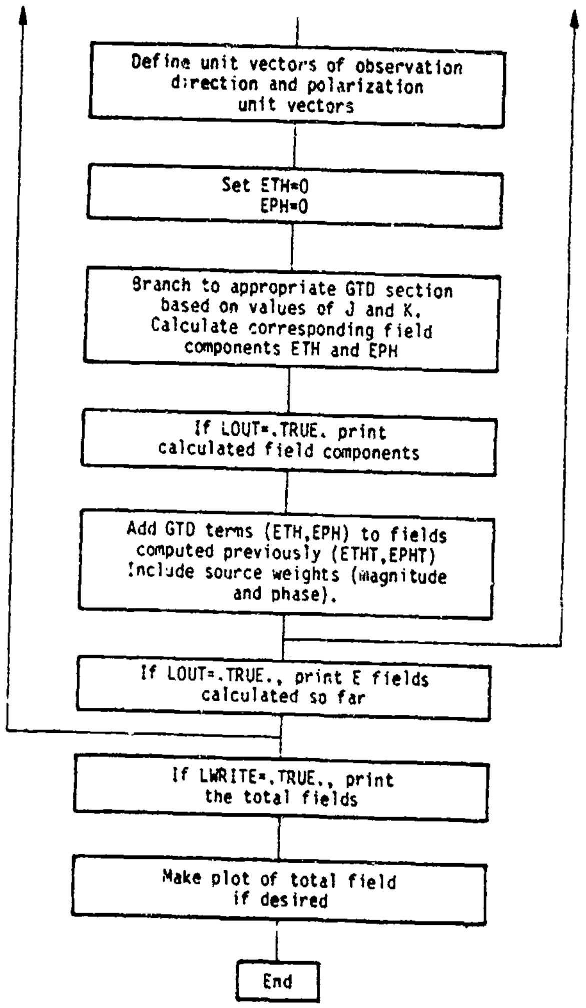
```
C!!!  2. INPUT CONVERSION SECTION
C!!!  NORMALIZE GEOMETRY UNITS (IN TERMS OF WAVELENGTHS) AND
C!!!  PERFORM RCS TRANSFORMATION (TO CYL COORD SYS) IF NEEDED
C!!!
C!!!  SET E FIELDS TO ZERO
      DO 1 I=1,361
      ETHT(I)=(Ø.,Ø.)
1     EPHT(I)=(Ø.,Ø.)
      FACTOR=1.
      BPL=Ø.
      BLR=BPL*RPD
      LNROT=.TRUE.
      DO 51Ø1 N=1,3
      XPC(N)=XPD(N)
      YPC(N)=YPD(N)
51Ø1  ZPC(N)=ZPD(N)
      IF(.NOT.LCYL) GO TO 4
      LRFC=.FALSE.
      IF(.NOT.LPLA) CO TO 51Ø6
      DO 51Ø5 MP=1,MPX
      LRFI(MP)=.FALSE.
      LRFS(MP)=.FALSE.
      MEX=MEP(MP)
      DO 51Ø5 ME=1,MEX
      LDRC(MP,ME)=.FALSE.
51Ø5  LRDC(MP,ME)=.FALSE.
51Ø6  CONTINUE
C!!!  DETERMINE IF RCS TRANSFORMATION IS NEEDED
      DO 8 N=1,3
      XCO(N)=XXCO(N)/WL
8     XOO(N)=Ø.
      LNROT=.TRUE.
      AXCL=ABS(XCL(1)-1.)
      AYCL=ABS(YCL(2)-1.)
      AZCL=ABS(ZCL(3)-1.)
      XCOM=SQRT(XCO(1)*XCO(1)+XCO(2)*XCO(2)+XCO(3)*XCO(3))
      IF(AXCL.GT.1.E-5.OR.AYCL.GT.1.E-5) LNROT=.FALSE.
      IF(AZCL.GT.1.E-5.OR.XCOM.GT.1.E-5) LNROT=.FALSE.
      IF(LNROT) GO TO 51ØØ
C!!!  REFER PATTERN AXES TO CYL. AXES.
      CALL ROTRAN(XPC,XPD,XOO)
      CALL ROTRAN(YPC,YPD,XCO)
      CALL ROTRAN(ZPC,ZPD,XOO)
51ØØ  CONTINUE
C!!!  NORMALIZE CYLINDER COORDINATES
      A=AA/WL
      B=BB/WL
      ZC(1)=ZCP/WL
      ZC(2)=ZCN/WL
      THTPR=THTP*RPD
      SNC(1)=SIN(THTPR)
      CNC(1)=COS(THTPR)
      CTC(1)=CNC(1)/SNC(1)
      THTNR=THTN*RPD
      SNC(2)=SIN(THTNR)
      CNC(2)=COS(THTNR)
      CTC(2)=CNC(2)/SNC(2)
4     CONTINUE
C!!!  NORMALIZE PLATE COORDINATES
      IF(.NOT.LPLA) GO TO 6
      DO 9 MP=1,MPXR
      MEX=MEP(MP)
      DO 9 ME=1,MEX
      DO 9 N=1,3
```

```
9     X(MP,ME,N)=X(MP,ME,N)/XL
      IF(LNROT) GO TO 5200
      DO 5210 MP=1,MPAR
      MEX=MEP(MP)
      DO 5210 ME=1,MEX
      DO 5220 N=1,3
5220  XXX(N)=X(MP,ME,N)
C!!!  REFER PLATE COORD. TO CYL. COORD SYS
      CALL ROTRAN(XXX,XXX,XCO)
      DO 5230 N=1,3
5230  X(MP,ME,N)=XXX(N)
5210  CONTINUE
5200  CONTINUE
6     CONTINUE
C!!!
```

### 3. Main Computation Section

This section directs the actual field calculations, performed in the various subroutines


Loop through sources
variable = MS
Specify source geometry
Perform normalization to wavelengths and transformation of source coordinates to cylinder coordinates if necessary
Define various geometrical properties of the structure
Initialize arrays used to define double diffraction sections
Loop through major GTD groupings
variable = K
K=1 calculate plate fields
K=2 calculate cylinder fields
K=3 calculate plate-cylinder interaction fields
Loop through individual GTD terms
variable = J
(see following sections)
Loop through pattern angle
Convert pattern angles to RCS


Define unit vectors of observation direction and polarization unit vectors
Set ETH=0 EPH=0
Branch to appropriate GTD section based on values of J and K. Calculate corresponding field components ETH and EPH
If LOUT=.TRUE. print calculated field components
Add GTD terms (ETH,EPH) to fields computed previously (ETHT,EPHT) Include source weights (magnitude and phase).
If LOUT=.TRUE., print E fields calculated so far
If LWRITE=.TRUE., print the total fields
Make plot of total field if desired
End

```
C!!!  3.  MAIN COMPUTATION SECTION
C!!!
C!!!  LOOP THRU VARIOUS SOURCES
      DO 1200 MS=1,MSX
C!!!  SPECIFY SOURCE GEOMETRY
C!!!  PERFORM NORMALIZATION AND TRANSFORMATION OF
C!!!  SOURCE COORDINATES
      DO 7 N=1,3
7     XS(N)=XSS(MS,N)/WL
      IM=IMS(MS)
      DO 5307 NJ=1,3
      DO 5307 NI=1,3
5307  VXS(NI,NJ)=VXSS(NI,NJ,MS)
      IF(LNROT) GO TO 5300
C!!!  REFER SOURCE LOCATION TO CYL. COORD SYS
      CALL ROTRAN(XS,XS,XCO)
C!!!  REFER SOURCE COORD SYS AXES TO CYL. COORD SYS
      DO 5304 NI=1,3
      DO 5303 NJ=1,3
5303  XXX(NJ)=VXSS(NI,NJ,MS)
      CALL ROTRAN(XXX,XXX,XCO)
      DO 5304 NJ=1,3
5304  VXS(NI,NJ)=XXX(NJ)
5300  CONTINUE
      IF(LAMP) GO TO 5301
      IF(IMNST.NE.0) GO TO 5305
C!!!  SPECIFY SOURCE DIMENSIONS
      HAW=HAWS(MS)
      H=HS(MS)
      GO TO 5306
5305  HAW=HAWS(MS)/WL
      H=HS(MS)/WL
5306  WI=WM(MS)*CEXP(CJ*WP(MS)*RPD)
      GO TO 5302
C!!!  SPECIFY SOURCE DIMENSIONS FOR NEC INPUT
5301  H=HS(MS)/WL
      HAW=0.
      WI=CMPLX(WM(MS),WP(MS))
      IF(H.LT.0.15) WI=0.5*PI*WI
5302  CONTINUE
C!!!  DEFINE VARIOUS GEOMETRY PROPERTIES OF STRUCTURE
      IF(LPLA) CALL GEOM
      IF(LCYL) CALL GEOMC
      IF(LPLA.AND.LCYL) CALL GEOMPC
C!!!
C!!!  NOTE:  AT THIS POINT THE RCS TRANSFORMATION (TO CYLINDER
C!!!         COORDINATES) IS COMPLETE.  THE CYLINDER COORD SYS
C!!!         AND RCS ARE NOW THE SAME AND WILL BE REFERRED TO
C!!!         AS THE RCS (REFERENCE COORD SYS)
C!!!
C!!!  INITIALIZE ARRAYS USED TO DEFINE DOUBLE DIFFRACTION SECTORS.
      DO 41 I=1,MEDX
      DO 41 J=1,MPDX
41    ID(J,I)=-1
      DO 42 I=1,361
42    IDD(I)=0
      KB=1
      KE=3
      IF(LSOR) GO TO 1148
      IF(.NOT.LCYL) GO TO 1148
      IF(LPLA) GO TO 1149
      KB=2
      KE=2
      GO TO 1149
1148  KB=1
      KE=1
```

```
1149  CONTINUE
C!!!  LOOP THRU MAJOR GTD GROUPS
C!!!     K=1   PLATE FIELDS
C!!!     K=2   CYLINDER FIELDS
C!!!     K=3   PLATE CYLINDER INTERACTION FIELDS
      DO 1150 K=KB,KE
      JB=JMN(K)
      JE=JMX(K)
      IF(LSOR) GO TO 1151
      IF(.NOT.LPLA.AND..NOT.LCYL) GO TO 1151
      IF(MPX.NE.0) GO TO 1152
      IF(K.EQ.2) GO TO 1152
      IF(JB.GT.2) GO TO 1150
      JE=2
      GO TO 1152
1151  JB=1
      JE=1
1152  CONTINUE
      IF(JB.EQ.0) GO TO 1150
      IF(JE.LT.JB) GO TO 1150
C!!!  LOOP THRU INDIVIDUAL GTD FIELDS.
      DO 1100 J=JB,JE
C!!!  LOOP THRU PATTERN ANGLE
      IF(LCNPAT) THP=TPPD
      IBP=IB+1
      IEP=IE+1
      IF(LDEBUG.OR.LTEST) IEP=IB+1
      DO 1100 II=IBP,IEP,IS
C!!!  CALCULATE PATTERN ANGLES IN PATTERN CUT COORD SYS.
      I=II-1
      PHP=I
      IF(LCNPAT) GO TO 1102
      IF(I.GT.180) GO TO 1101
      PHP=TPPD
      THP=I
      GO TO 1102
1101  PHP=TPPD+180.
      IF(PHP.GE.360.) PHP=PHP-360.
      THP=360-I
1102  THPR=THP*RPD
      PHPR=PHP*RPD
C!!!  CONVERT PATTERN ANGLES TO REF. COORD. SYS.
      CALL PATROT(THSR,PHSR,THPR,PHPR,ALR)
      STHS=SIN(THSR)
      CTHS=COS(THSR)
      SPS=SIN(PHSR)
      CPS=COS(PHSR)
      AS=PI-THSR
      SAS=SIN(AS)
      SASP=ABS(SIN(AS-0.5*PI))
      CAS=COS(AS)
C!!!  DEFINE OBSERVATION DIRECTION AND THETA,PHI UNIT VECTORS.
      D(1)=STHS*CPS
      D(2)=STHS*SPS
      D(3)=CTHS
      DT(1)=CTHS*CPS
      DT(2)=CTHS*SPS
      DT(3)=-STHS
      DP(1)=-SPS
      DP(2)=CPS
      ETH=(0.,0.)
      EPH=(0.,0.)
C!!!  BRANCH TO APPROPRIATE GTD SECTION BASED ON VALUES OF J AND K
      GO TO (1110,1120,1130),K
1110  CONTINUE
      GO TO (100,200,300,600,700,800,900),J
```

```
 100   CONTINUE
C!!!   COMPUTE THE DIRECT FIELD FROM THE SOURCE.
       CALL INCFLD(EITH,EIPH,LSOR)
       ETH=EITH
       EPH=EIPH
       IF(LOUT) CALL PRIOUT(100,0,0,0,EITH,EIPH)
       GO TO 1000
 200   CONTINUE
C!!!   COMPUTE ALL POSSIBLE SINGLY REFLECTED FIELDS FROM PLATES.
       DO 25 MP=1,MPXR
C!!!    IF SLOT ON PLATE,THEN NO REFL. FIELD.
       IF(LSURF(MP)) GO TO 25
C!!!    IF PLATE SHADOWED,THEN NO REFL. FIELD.
       IF(LSHD(MP)) GO TO 25
       CALL REFPLA(ERPTH,ERPPH,MP)
       ETH=ETH+ERPTH
       EPH=EPH+ERPPH
       IF(LOUT) CALL PRIOUT(200,MP,0,0,ERPTH,ERPPH)
 25    CONTINUE
       GO TO 1000
 300   CONTINUE
C!!!   COMPUTE ALL POSSIBLE DOUBLY REFLECTED FIELDS.
       DO 31 MP=1,MPXR
C!!!    IF SLOT ON PLATE,THEN NO REFL/REFL FIELD.
       IF(LSURF(MP)) GO TO 31
C!!!    IF PLATE #MP IS SHADOWED,THEN NO REFL. FIELD
       IF (LSHD(MP)) GO TO 31
       DO 30 MPP=1,MPXR
        IF (MPP.EQ.MP) GO TO 30
       IF(LIHD(MP,MPP))GO TO 30
       CALL RPLRPL(ERRPT,ERRPP,MP,MPP)
       ETH=ETH+ERRPT
       EPH=EPH+ERRPP
       IF(LOUT) CALL PRIOUT(300,MP,MPP,0,ERRPT,ERRPP)
 30    CONTINUE
 31    CONTINUE
       GO TO 1000
 600   CONTINUE
C!!!   COMPUTE ALL POSSIBLE SINGLY DIFFRACTED FIELDS INCLUDE
C!!!   A CORNER DIFFRACTION TERM IF DESIRED BY INPUT DATA.
       DO 61 MP=1,MPX
C!!!    IF PLATE SHADOWED,THEN NO DIFF. FIELD.
       IF(LSHD(MP)) GO TO 61
       MEX=MEP(MP)
       DO 60 ME=1,MEX
       FN=FNP(MP,ME)
C!!!    IF WEDGE ANGLE INDICATOR (FN)<0,THEN HAVE COMMON EDGE ON
C!!!    OTHER PLATE COMPUTE DIFF. FIELD.
       IF(FN.LT.0.) GO TO 60
       CALL DIFPLT(EDPTH,EDPPH,EDPCTH,EDPCPH,FN,ME,MP)
       ETH=ETH+EDPTH+EDPCTH
       EPH=EPH+EDPPH+EDPCPH
       IF(LOUT) CALL PRIOUT(600,MP,ME,0,EDPTH,EDPPH)
       IF (LOUT) CALL PRIOUT(650,MP,ME,0,EDPCTH,EDPCPH)
 60    CONTINUE
 61    CONTINUE
       GO TO 1000
 700   CONTINUE
C!!!   LOOP THRU THE VARIOUS REFLECTED/DIFFRACTED FIELDS.
C!!!   INCLUDE CORNER TERM IF DESIRED BY INPUT DATA.
       DO 72 MR=1,MPXR
C!!!    IF SLOT ON PLATE,THEN NO REFL/DIFF FIELD.
       IF(LSURF(MR)) GO TO 72
C!!!    IF PLATE #MR IS SHADOWED,THEN NO REFL. FIELD
       IF (LSHD(MR)) GO TO 72
       DO 71 MP=1,MPX
```

```
      IF (MP.EQ.MR) GO TO 71
      IF(LIHD(MR,MP))GO TO 71
      MEX=MEP(MP)
      DO 70 ME=1,MEX
      FN=FNP(MP,ME)
C!!!  IF FN<0 THEN HAVE COMMON EDGE ON
C!!!  OTHER PLATE COMPUTE DIFF. FIELD.
      IF(FN.LT.0.) GO TO 70
      CALL RFLDPL(ERPDT,ERPDP,ERPDCT,ERPDCP,FN,ME,MP,MR)
      ETH=ETH+ERPDT+ERPDCT
      EPH=EPH+ERPDP+ERPDCP
      IF(LOUT) CALL PRIOUT(700,MR,MP,ME,ERPDT,ERPDP)
      IF (LOUT) CALL PRIOUT(750,MR,MP,ME,ERPDCT,ERPDCP)
 70   CONTINUE
 71   CONTINUE
 72   CONTINUE
      GO TO 1000
 800  CONTINUE
C!!!  COMPUTE THE VARIOUS DIFFRACTED/REFLECTED FIELDS.
C!!!  INCLUDE CORNER TERM IF DESIRED BY INPUT DATA.
      DO 82 MP=1,MPX
C!!!  IF PLATE IS SHADOWED,THEN NO DIFF/REFL FIELD.
      IF(LSHD(MP)) GO TO 82
      MEX=MEP(MP)
      DO 81 ME=1,MEX
      FN=FNP(MP,ME)
      IF(FN.LT.0.) GO TO 81
      DO 80 MR=1,MPXR
      IF(MR.EQ.MP) GO TO 80
      IF(LIHD(MP,MR))GO TO 80
      CALL DPLRPL(EDRPT,EDRPP,EDCRPT,EDCRPP,FN,ME,MP,MR)
      ETH=ETH+EDRPT+EDCRPT
      EPH=EPH+EDRPP+EDCRPP
      IF(LOUT) CALL PRIOUT(800,MP,ME,MR,EDRPT,EDRPP)
      IF (LOUT) CALL PRIOUT (850,MP,ME,MR,EDCRPT,EDCRPP)
 80   CONTINUE
 81   CONTINUE
 82   CONTINUE
      GO TO 1000
 900  CONTINUE
C!!!  CHECK TO SEE IF DOUBLE DIFFRACTIONS OCCUR.
C!!!  IF SO, INDICATE IN OUTPUT FILE.
      IF(IDD(II).GE.0)GO TO 911
      ME=-IDD(II)/400
      MP=-IDD(II)/20-ME*20
      MPP=-IDD(II)-ME*400-MP*20
      IF(LCRNR.AND.MPP.GE.MPXR) GO TO 911
      IF(MPP.EQ.0) GO TO 912
      WRITE(6,913)I,MP,ME,MPP
 913  FORMAT(' DOUBLE DIFFRACTION AT ANGLE= ',I3,' FROM PLATE# '
     2,I2,' EDGE# ',I2,' IS SHADOWED BY PLATE# ',I2)
      GO TO 911
 912  WRITE(6,914) I,MP,ME
 914  FORMAT(' DOUBLE DIFFRACTION AT ANGLE= ',I3,' FROM PLATE# '
     2,I2,' EDGE# ',I2,' IS SHADOWED BY THE CYLINDER')
 911  CONTINUE
      GO TO 1000
 1120 CONTINUE
      GO TO (101,150,500),J
 101  CONTINUE
C!!!  COMPUTE DIRECT FIELD FROM SOURCE
      IF(LPLA) GO TO 12
      CALL INCFLD(EITH,EIPH,LSOR)
      ETH=EITH
      EPH=EIPH
      IF(LSOR) GO TO 1000
```

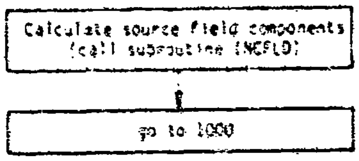
```
12     CONTINUE
C!!!   COMPUTE SCATTERED FIELD FROM CYLINDEP
       CALL SCTCYL(ESTH,ESPH,ERTH,ERPH)
       ETH=ETH+ESTH
       EPH=EPH+ESPH
       IF(.NOT.LOUT) GO TO 1ØØØ
       CALL PRIOUT(11Ø,Ø,Ø.Ø,EITH,EIPH)
       CALL PRIOUT(12Ø,Ø,Ø,Ø,ERTH,ERPH)
       CALL PRIOUT(13Ø,Ø,Ø,Ø,ESTH,ESPH)
       GO TO 1ØØØ
15Ø    CONTINUE
C!!!   COMPUTE ALL POSSIBLE REFLECTED FIELDS FROM END CAPS
       DO 15 MC=1,2
C!!! IF ANTENNA IS ON ENDCAP NO REFLECTED FIELD FROM END CAP
       IF(LSRFC(MC)) GO TO 15
       CALL REFCAP(ERCAT,ERCAP,MC)
       ETH=ETH+ERCAT
       EPH=EPH+ERCAP
       IF(LOUT) CALL PRIOUT(15Ø,MC,Ø,Ø,ERCAT.ERCAP)
15     CONTINUE
       GO TO 1ØØØ
5ØØ    CONTINUE
C!!!   COMPUTE ALL POSSIBLE DIFFRACTED FIELDS FROM END CAPS
       DO 5Ø MC=1,2
       CALL ENDIF(EDCTH,EDCPH,MC)
       ETH=ETH+EDCTH
       EPH=EPH+EDCPH
       IF(LOUT) CALL PRIOUT(5ØØ,MC,Ø,Ø,EDCTH,EDCPH)
5Ø     CONTINUE
       GO TO 1ØØØ
13Ø    CONTINUE
       GO TO (25Ø,4ØØ,94Ø,95Ø),J
25Ø    CONTINUE
C!!!   COMPUTE ALL POSSIBLE FIELDS REFLECTED FROM THE PLATES THEN
C!!!   SCATTERED FROM THE CYLINDER .
       DO 29 MP=1,MPXR
C!!!   IF ANTENNA IS ON PLATE, THEN NO REFLECTED FIELD
       IF(LSURF(MP)) GO TO 29
C!!!   IF PLATE SHADOWED, THEN NO REFLECTED FIELD
       IF(LSHD(MP)) GO TO 29
       CALL RPLSCL(ERPST,ERPSP,ERPCT,ERPCP,MP)
       ETH=ETH+ERPST
       EPH=EPH+ERPSP
       IF(.NOT.LOUT) GO TO 29
       CALL PRIOUT(24Ø,MP,Ø,Ø,ERPCT,ERPCP)
       CALL PRIOUT(25Ø,MP,Ø,Ø,ERPST,ERPSP)
29     CONTINUE
       GO TO 1ØØØ
4ØØ    CONTINUE
C!!!   COMPUTE ALL POSSIBLE FIELDS SCATTERED FROM THE CYLINDER THEN
C!!!   REFLECTED FROM THE PLATES
       DO 4Ø MP=1,MPXR
       CALL SCLRPL(ERSPT,ERSPP.ERCPT,ERCPP,MP)
       ETH=ETH+ERSPT
       EPH=EPH+ERSPP
       IF(.NOT.LOUT) GO TO 4Ø
       CALL PRIOUT(41Ø,MP,Ø,Ø,ERCPT,ERCPP)
       CALL PRIOUT(42Ø,MP,Ø,Ø,ERSPT,ERSPP)
4Ø     CONTINUE
       GO TO 1ØØØ
94Ø    CONTINUE
C!!!   COMPUTE ALL POSSIBLE FIELDS REFLECTED FROM THE CYLINDER THEN
C!!!   DIFFRACTED FROM THE PLATES
       DO 91 MP=1,MPX
C!!!   IF PLATE SHADOWED, THEN NO DIFFRACTED FIELD
       IF(LSHD(MP)) GO TO 91
```

```
      MEX=MEP(MP)
      DO 9Ø ME=1,MEX
      FN=FNP(MP,ME)
      IF(FN.LT.Ø.) GO TO 9Ø
      CALL RCLDPL(ERDTH,ERDPH,FN,ME,MP)
      ETH=ETH+ERDTH
      EPH=EPH+ERDPH
      IF(LOUT) CALL PRIOUT(94Ø,MP,ME,Ø,ERDTH,ERDPH)
 9Ø   CONTINUE
 91   CONTINUE
      GO TO 1ØØØ
 95Ø  CONTINUE
C!!!  COMPUTE ALL POSSIBLE FIELDS DIFFRACTED FROM THE PLATES THEN
C!!!  REFLECTED FROM THE CYLINDER
      DO 96 MP=1,MPX
C!!!  IF PLATE SHADOWED, THEN NO DIFFRACTED FIELD
      IF(LSHD(MP)) GO TO 96
      MEX=MEP(MP)
      DO 95 ME=1,MEX
C!!!  IF EDGE DOES NOT HAVE STRONG FIELD REFLECTED FROM CYLINDER
C!!!  BYPASS SUBR.
      IF(.NOT.LDC(MP,ME)) GO TO 95
      FN=FNP(MP,ME)
      IF(FN.LT.Ø.) GO TO 95
      CALL DPLRCL(EDRCT,EDRCP,FN,ME,MP)
      ETH=ETH+EDRCT
      EPH=EPH+EDRCP
      IF(LOUT) CALL PRIOUT(95Ø,MP,ME,Ø,EDRCT,EDRCP)
 95   CONTINUE
 96   CONTINUE
 1ØØØ CONTINUE
      IF(LOUT) CALL PRIOUT(I,I,J,J,ETH,EPH)
C!!!  SUPERPOSITION OF THE FIELD COMPONENTS, WEIGHTING
C!!!  OF RESULT IN TERMS OF THE INPUT EXCITATION, AND
C!!!  THE CONVERSION OF THE POLARIZATION TO THE
C!!!  PATTERN CUT COORDINATE SYSTEM.
      ETHT(II)=ETHT(II)+WI*(ETH*COS(ALR+BLR)+EPH*SIN(ALR+BLR))
      EPHT(II)=EPHT(II)+WI*(EPH*COS(ALR+BLR)-ETH*SIN(ALR+BLR))
 11ØØ CONTINUE
 115Ø CONTINUE
      IF (.NOT.LOUT) GO TO 12ØØ
      DO 12Ø2 II=IEP,IEP,IS
      I=II-1
 12Ø2 CALL PRIOUT (1ØØØ,I,I,I,ETHT(II),EPHT(II))
 12ØØ CONTINUE
C!!!  E-THETA AND E-PHI RESULTS ARE SENT TO UNIT #6(LINE PRINTER).
      IEE=IEP-1
      IF(LWRITE) CALL OUTPUT(ETHT,EPHT,LCMPAT,TPPD,IB,IEE,IS)
C!!!  POLAR PLOT OF DATA IF DESIRED.
C!!!  NOTE THAT THE PLOT ROUTINES ARE NOT INCLUDED
C!!!  SINCE THEY CAN NOT BE USED ON ALL SYSTEMS.
      IF(.NOT.LPLT) GO TO 998
C ADD CALL POLPLT(ETHT,RADIUS,IPLT,IS,361)
C ADD CALL POLPLT(EPHT,RADIUS,IPLT,IS,361)
 998  CONTINUE
      GO TO 2999
 999  STOP
      END
```

A description of the various GTD computation sections based on values of J and K follows. A partial listing of each section is repeated for clarity.

K=1, J=1

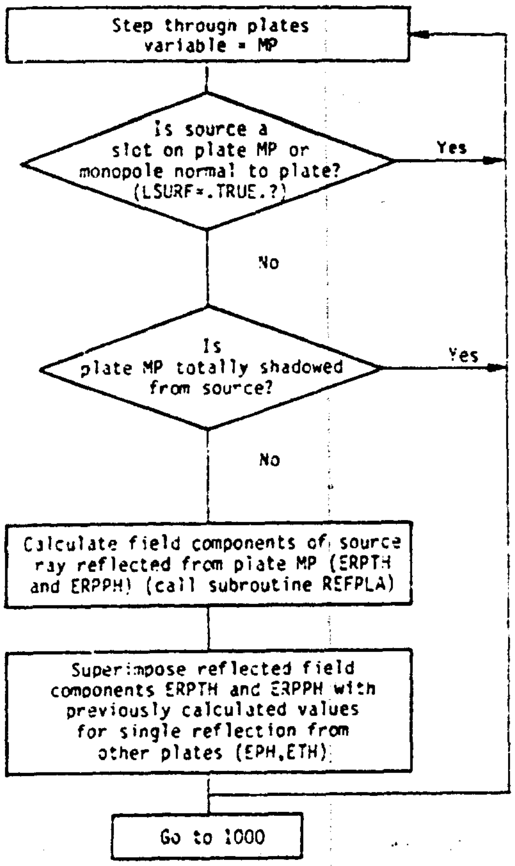
This section calculates the geometrical optics source field.



```
 100   CONTINUE
C!!!   COMPUTE THE DIRECT FIELD FROM THE SOURCE.
       CALL INCFLD(EITH,EIPH,LSOR)
       ETH=EITH
       EPH=EIPH
       IF(LOUT) CALL PRIOUT(100,0,0,0,EITH,EIPH)
       GO TO 1000
```

K=1,J=2

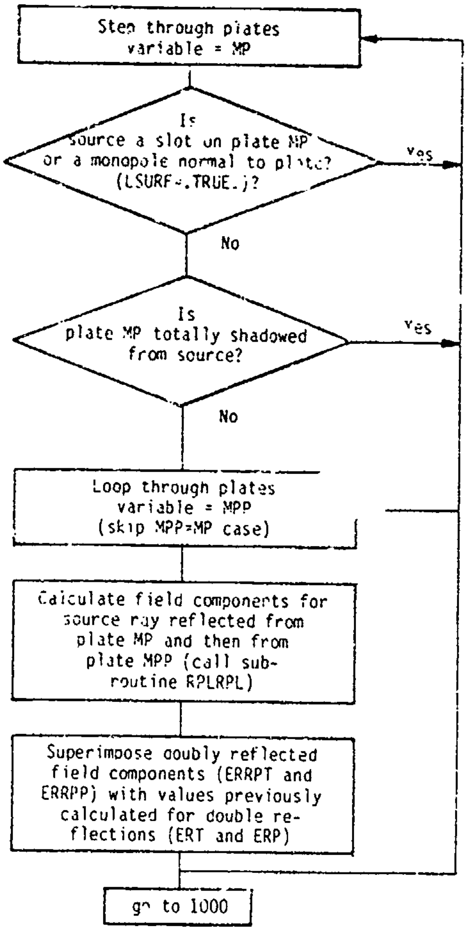
This section calculates all fields singly reflected from plates.


Step through plates variable = MP
Is source a slot on plate MP or monopole normal to plate? (LSURF=.TRUE.?)
Yes
No
Is plate MP totally shadowed from source?
Yes
No
Calculate field components of source ray reflected from plate MP (ERPTH and ERPPH) (call subroutine REFPLA)
Superimpose reflected field components ERPTH and ERPPH with previously calculated values for single reflection from other plates (EPH,ETH)
Go to 1000

```
200   CONTINUE
C!!!  COMPUTE ALL POSSIBLE SINGLY REFLECTED FIELDS FROM PLATES.
      DO 25 MP=1,MPXR
C!!!  IF SLOT ON PLATE,THEN NO REFL. FIELD.
      IF(LSURF(MP)) GO TO 25
C!!!  IF PLATE SHADOWED,THEN NO REFL. FIELD.
      IF(LSHD(MP)) GO TO 25
      CALL REFPLA(ERPTH,ERPPH,MP)
      ETH=ETH+ERPTH
      EPH=EPH+ERPPH
      IF(LOUT) CALL PRIOUT(200,MP,0,0,ERPTH,ERPPH)
25    CONTINUE
      GO TO 1000
```

K 1, J-3

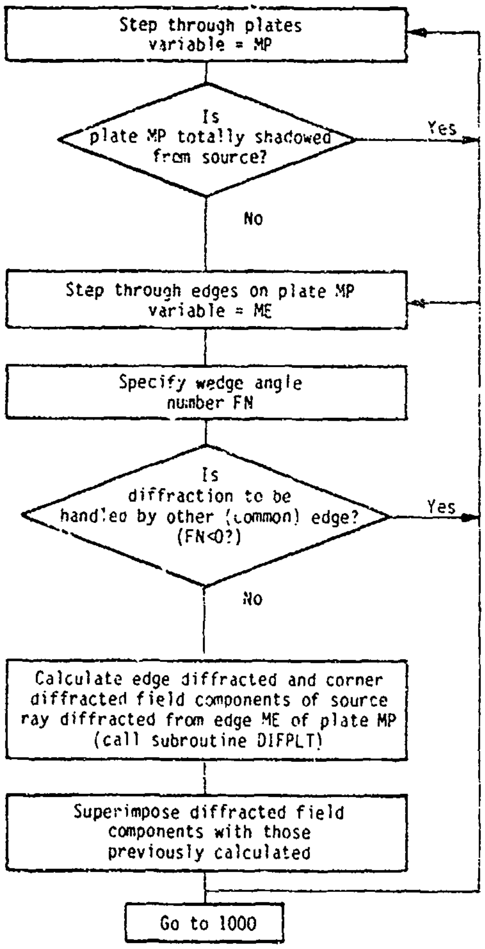
This section computes all possible doubly reflected fields from plates.


Step through plates
variable = MP
Is source a slot on plate MP or a monopole normal to plate? (LSURF=.TRUE.)?
Yes
No
Is plate MP totally shadowed from source?
Yes
No
Loop through plates
variable = MPP
(skip MPP=MP case)
Calculate field components for source ray reflected from plate MP and then from plate MPP (call subroutine RPLRPL)
Superimpose doubly reflected field components (ERRPT and ERRPP) with values previously calculated for double reflections (ERT and ERP)
go to 1000

```
 300   CONTINUE
 C!!!  COMPUTE ALL POSSIBLE DOUBLY REFLECTED FIELDS.
       DO 31 MP=1,MPXR
 C!!!  IF SLOT ON PLATE,THEN NO REFL/REFL FIELD.
       IF(LSURF(MP)) GO TO 31
 C!!!  IF PLATE #MP IS SHADOWED,THEN NO REFL. FIELD
       IF (LSHD(MP)) GO TO 31
       DO 30 MPP=1,MPXR
       IF (MPP.EQ.MP) GO TO 30
       IF(LIHD(MP,MPP))GO TO 30
       CALL RPLRPL(ERRPT,ERRPP,MP,MPP)
       ETH=ETH+ERRPT
       EPH=EPH+ERRPP
       IF(LOUT) CALL PRIOUT(300,MP,MPP,0,ERRPT,ERRPP)
 30    CONTINUE
 31    CONTINUE
       GO TO 1000
```

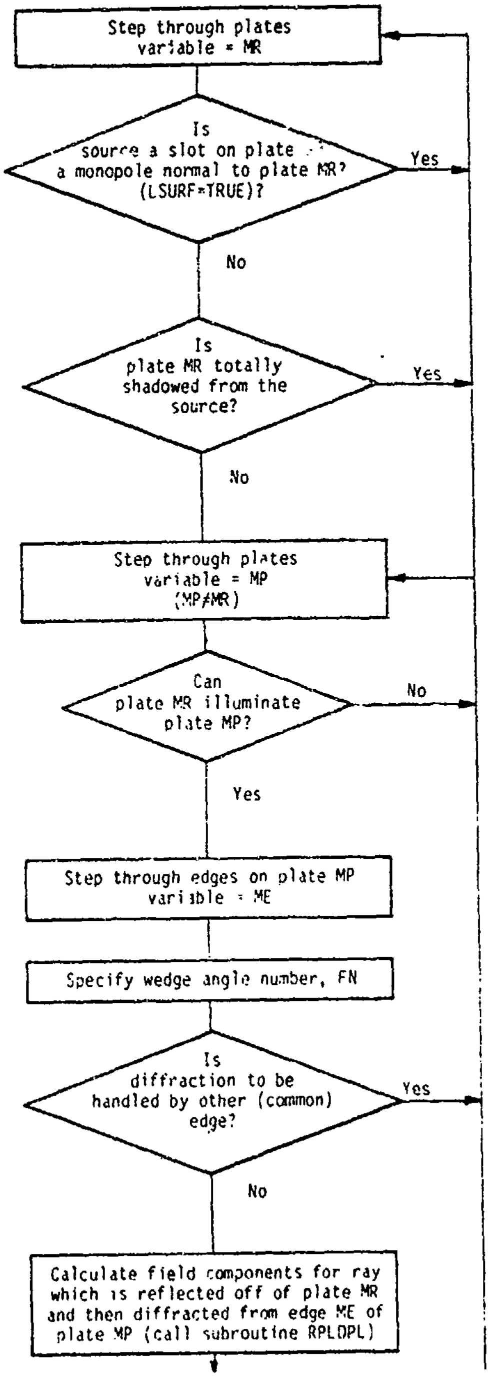
K=1, J=4

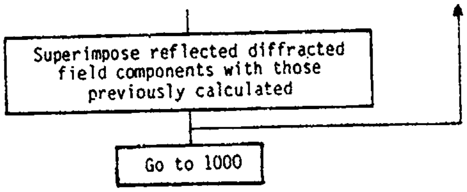
This section computes field components for all source rays singly diffracted by plate edges.


Step through plates
variable = MP
Is
plate MP totally shadowed
from source?
Yes
No
Step through edges on plate MP
variable = ME
Specify wedge angle
number FN
Is
diffraction to be
handled by other (common) edge?
(FN<0?)
Yes
No
Calculate edge diffracted and corner
diffracted field components of source
ray diffracted from edge ME of plate MP
(call subroutine DIFPLT)
Superimpose diffracted field
components with those
previously calculated
Go to 1000

```
600   CONTINUE
C!!!  COMPUTE ALL POSSIBLE SINGLY DIFFRACTED FIELDS INCLUDE
C!!!  A CORNER DIFFRACTION TERM IF DESIRED BY INPUT DATA.
      DO 61 MP=1,MPX
C!!!  IF PLATE SHADOWED,THEN NO DIFF. FIELD.
      IF(LSHD(MP)) GO TO 61
      MEX=MEP(MP)
      DO 60 ME=1,MEX
      FN=FNP(MP,ME)
C!!!  IF WEDGE ANGLE INDICATOR (FN)<0,THEN HAVE COMMON EDGE ON
C!!!  OTHER PLATE COMPUTE DIFF. FIELD.
      IF(FN.LT.0.) GO TO 60
      CALL DIFPLT(EDPTH,EDPPH,EDPCTH,EDPCPH,FN,ME,MP)
      ETH=ETH+EDPTH+EDPCTH
      EPH=EPH+EDPPH+EDPCPH
      IF(LOUT) CALL PRIOUT(600,MP,ME,0,EDPTH,EDPPH)
      IF (LOUT) CALL PRIOUT(650,MP,ME,0,EDPCTH,EDPCPH)
60    CONTINUE
61    CONTINUE
      GO TO 1600
```

K=1, J=5
This section computes field components for all source rays reflected by a plate and then diffracted from an edge on another plate.

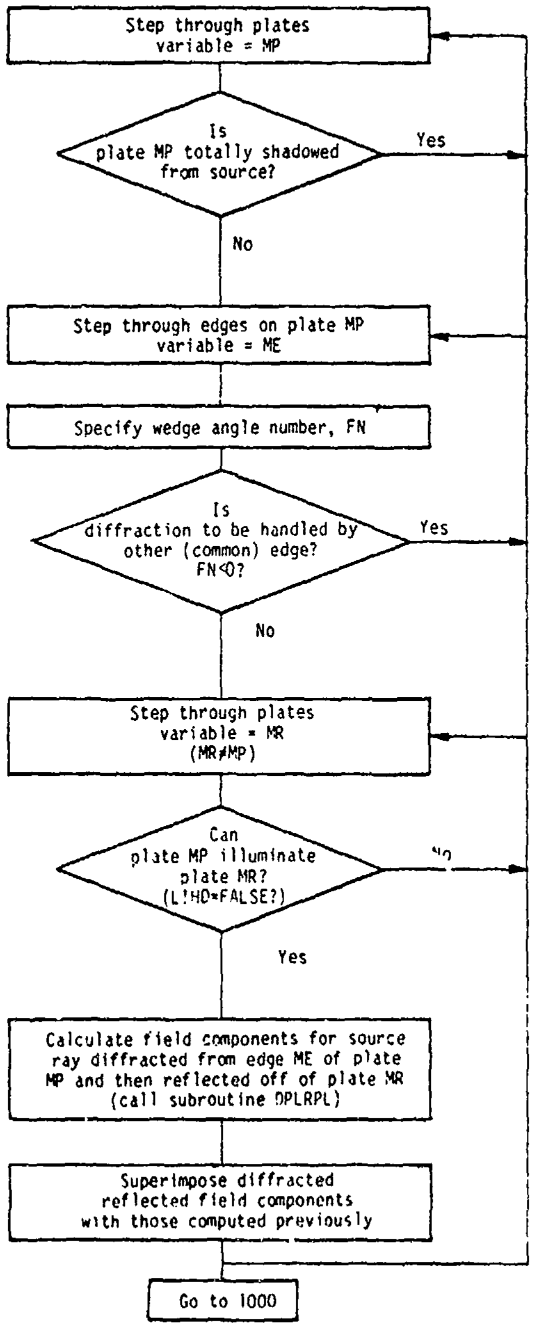

Step through plates
variable = MR
Is
source a slot on plate
a monopole normal to plate MR?
(LSURF=TRUE)?
Yes
No
Is
plate MR totally
shadowed from the
source?
Yes
No
Step through plates
variable = MP
(MP≠MR)
Can
plate MR illuminate
plate MP?
No
Yes
Step through edges on plate MP
variable = ME
Specify wedge angle number, FN
Is
diffraction to be
handled by other (common)
edge?
Yes
No
Calculate field components for ray
which is reflected off of plate MR
and then diffracted from edge ME of
plate MP (call subroutine RPLDPL)


Superimpose reflected diffracted field components with those previously calculated
Go to 1000

```
 700    CONTINUE
C!!!    LOOP THRU THE VARIOUS REFLECTED/DIFFRACTED FIELD TERMS.
C!!!    INCLUDE CORNER TERM IF DESIRED BY INPUT DATA.
        DO 72 MR=1,MPXR
C!!!    IF SLOT ON PLATE,THEN NO REFL/DIFF FIELD.
        IF(LSURF(MR)) GO TO 72
C!!!    IF PLATE #MR IS SHADOWED,THEN NO REFL. FIELD
        IF (LSHD(MR)) GO TO 72
        DO 71 MP=1,MPX
        IF (MP.EQ.MR) GO TO 71
        IF(LIHD(MR,MP))GO TO 71
        MEX=MEP(MP)
        DO 70 ME=1,MEX
        FN=FNP(MP,ME)
C!!!    IF FN<0 THEN HAVE COMMON EDGE ON
C!!!    OTHER PLATE COMPUTE DIFF. FIELD.
        IF(FN.LT.0.) GO TO 70
        CALL RPLDPL(ERPDT,ERPDP,ERPDCT,ERPDCP,FN,ME,MP,MR)
        ETH=ETH+ERPDT+ERPDCT
        EPH=EPH+ERPDP+ERPDCP
        IF(LOUT) CALL PRIOUT(700,MR,MP,ME,ERPDT,ERPDP)
        IF (LOUT) CALL PRIOUT(750,MR,MP,ME,ERPDCT,ERPDCP)
 70     CONTINUE
 71     CONTINUE
 72     CONTINUE
        GO TO 1000
```

K=1 J=6

This section computes field components for all source rays diffracted from a plate edge and then reflected off of another plate.

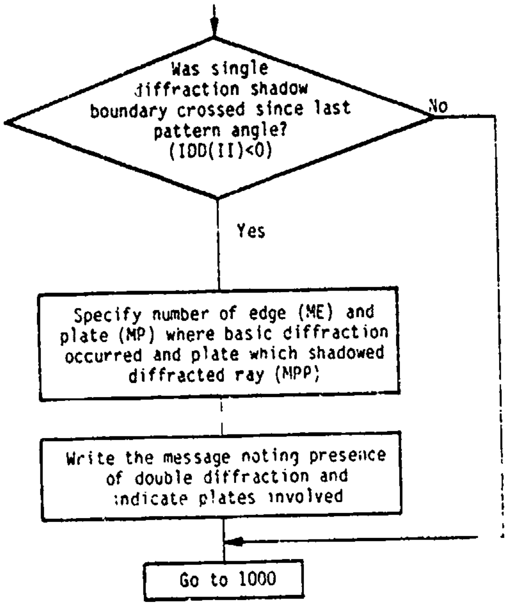


Step through plates
variable = MP
Is
plate MP totally shadowed
from source?
Yes
No
Step through edges on plate MP
variable = ME
Specify wedge angle number, FN
Is
diffraction to be handled by
other (common) edge?
FN<0?
Yes
No
Step through plates
variable = MR
(MR≠MP)
Can
plate MP illuminate
plate MR?
(L!HD=FALSE?)
No
Yes
Calculate field components for source
ray diffracted from edge ME of plate
MP and then reflected off of plate MR
(call subroutine DPLRPL)
Superimpose diffracted
reflected field components
with those computed previously
Go to 1000

```
600   CONTINUE
C!!!  COMPUTE THE VARIOUS DIFFRACTED/REFLECTED FIELDS.
C!!!  INCLUDE CORNER TERM IF DESIRED BY INPUT DATA.
      DO 82 MP=1,MPX
C!!!  IF PLATE IS SHADOWED,THEN NO DIFF/REFL FIELD.
      IF(LSHD(MP)) GO TO 82
      MEX=MEP(MP)
      DO 81 ME=1,MEX
      FN=FNP(MP,ME)
      IF(FN.LT.Ø.) GO TO 81
      DO 8Ø MR=1,MPXR
      IF(MR.EQ.MP) GO TO 8Ø
      IF(LIHD(MP,MR))GO TO 8Ø
      CALL DPLRPL(EDRPT,EDRPP,EDCRPT,EDCRPP,FN,ME,MP,MR)
      ETH=ETH+EDRPT+EDCRPT
      EPH=EPH+EDRPP+EDCRPP
      IF(LOUT) CALL PRIOUT(8ØØ,MP,ME,MR,EDRPT,EDRPP)
      IF (LOUT) CALL PRIOUT (85Ø,MP,ME,MR,EDCRPT,EDCRPP)
8Ø    CONTINUE
81    CONTINUE
82    CONTINUE
      GO TO 16ØØ
```

K=1, J=7

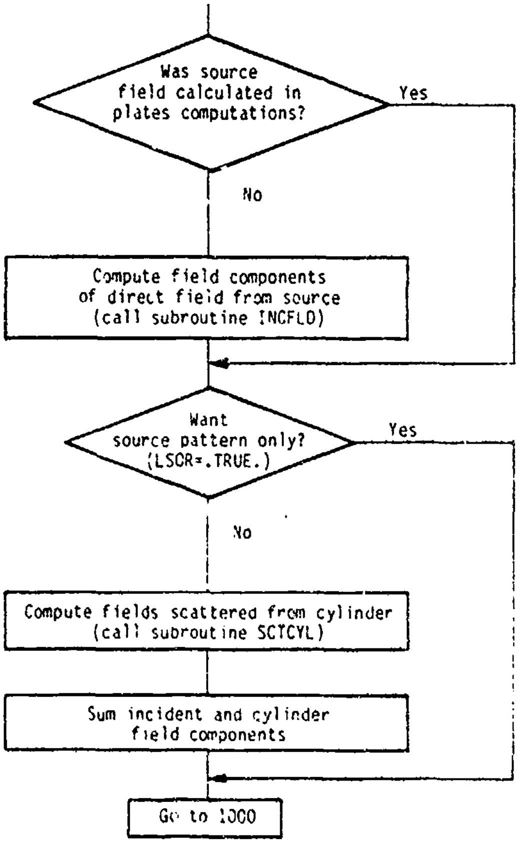
This section identifies double diffraction shadow boundaries.


Was single diffraction shadow boundary crossed since last pattern angle? (IDD(II)<0)
No
Yes
Specify number of edge (ME) and plate (MP) where basic diffraction occurred and plate which shadowed diffracted ray (MPP)
Write the message noting presence of double diffraction and indicate plates involved
Go to 1000

```
 900   CONTINUE
C!!!   CHECK TO SEE IF DOUBLE DIFFRACTIONS OCCUR.
C!!!   IF SO, INDICATE IN OUTPUT FILE.
       IF(IDD(II).GE.0)GO TO 911
       ME=-IDD(II)/400
       MP=-IDD(II)/20-ME*20
       MPP=-IDD(II)-ME*400-MP*20
       IF(LGRND.AND.MPP.GE.MPXR) GO TO 911
       IF(MPP.EQ.0) GO TO 912
       WRITE(6,913)I,MP,ME,MPP
 913   FORMAT(' DOUBLE DIFFRACTION AT ANGLE= ',I3,' FROM PLATE# '
      2,I2,' EDGE# ',I2,' IS SHADOWED BY PLATE# ',I2)
       GO TO 911
 912   WRITE(6,914) I,MP,ME
 914   FORMAT(' DOUBLE DIFFRACTION AT ANGLE= ',I3,' FROM PLATE# '
      2,I2,' EDGE# ',I2,' IS SHADOWED BY THE CYLINDER')
 911   CONTINUE
       GO TO 1000
```

K=2, J=1

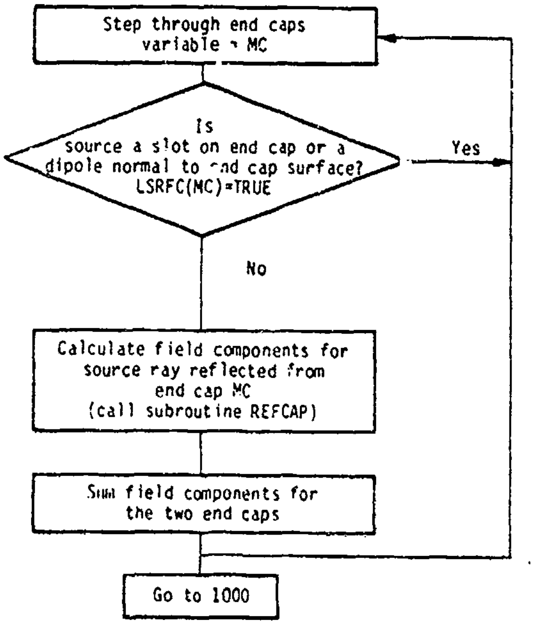
This section computes the source field and the field scattered from the cylinder.


Was source field calculated in plates computations?
Yes
No
Compute field components of direct field from source (call subroutine INCFLD)
Want source pattern only? (LSCR=.TRUE.)
Yes
No
Compute fields scattered from cylinder (call subroutine SCTCYL)
Sum incident and cylinder field components
Go to 1000

```
101    CONTINUE
C!!!   COMPUTE DIRECT FIELD FROM SOURCE
       IF(LPLA) GO TO 12
       CALL INCFLD(EITH,EIPH,LSCR)
       ETH=EITH
       EPH=EIPH
       IF(LSOR) GO TO 1000
12     CONTINUE
C!!!   COMPUTE SCATTERED FIELD FROM CYLINDER
       CALL SCTCYL(ESTH,ESPH,ERTH,ERPH)
       ETH=ETH+ESTH
       EPH=EPH+ESPH
       IF(.NOT.LOUT) GO TO 1000
       CALL PRIOUT(110,0,0,0,EITH,EIPH)
       CALL PRIOUT(120,0,0,0,ERTH,ERPH)
       CALL PRIOUT(130,0,0,0,ESTH,ESPH)
       GO TO 1000
```

K=2, J=2

This section computes fields reflected from cylinder end caps.

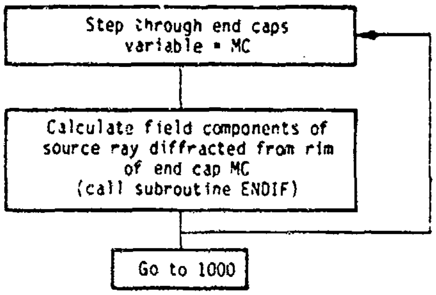


Step through end caps variable → MC
Is source a slot on end cap or a dipole normal to end cap surface? LSRFC(MC)=TRUE
Yes
No
Calculate field components for source ray reflected from end cap MC (call subroutine REFCAP)
Sum field components for the two end caps
Go to 1000

```
150   CONTINUE
C!!!  COMPUTE ALL POSSIBLE REFLECTED FIELDS FROM END CAPS
      DO 15 MC=1,2
C!!! IF ANTENNA IS ON ENDCAP NO REFLECTED FIELD FROM END CAP
      IF(LSRFC(MC)) GO TO 15
      CALL REFCAP(ERCAT,ERCAP,MC)
      ETH=ETH+ERCAT
      EPH=EPH+ERCAP
      IF(LOUT) CALL PRIOUT(150,MC,Ø,Ø,ERCAT,ERCAP)
15    CONTINUE
      GO TO 1ØØØ
```

K=2, J=3

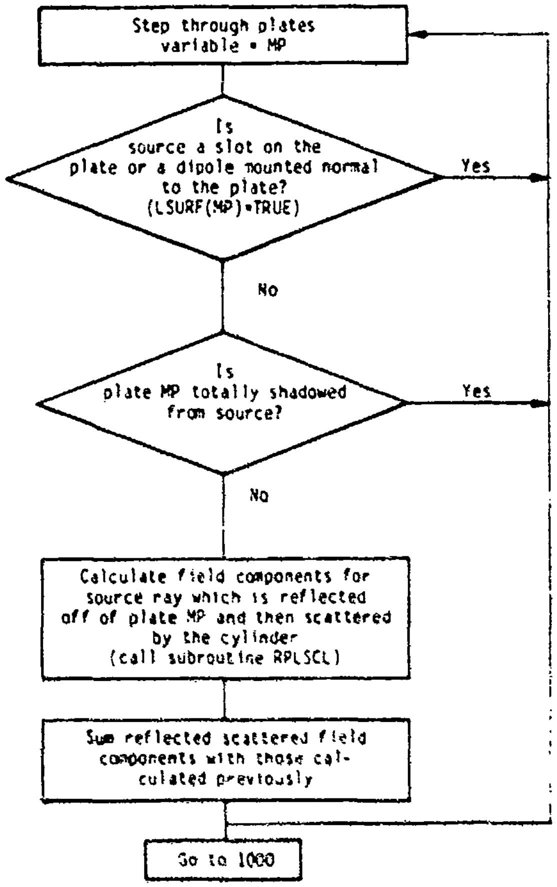
This section computes field components for all source rays diffracted from the cylinder end cap rims.



```
500   CONTINUE
C***  COMPUTE ALL POSSIBLE DIFFRACTED FIELDS FROM END CAPS
      DO 50 MC=1,2
      CALL ENDIF(EDCTH,EDCPH,MC)
      ETH=ETH+EDCTH
      EPH=EPH+EDCPH
      IF(LOUT) CALL PRIOUT(500,MC,Ø,Ø,EDCTH,EDCPH)
50    CONTINUE
      GO TO 1000
```

K=3, J=1

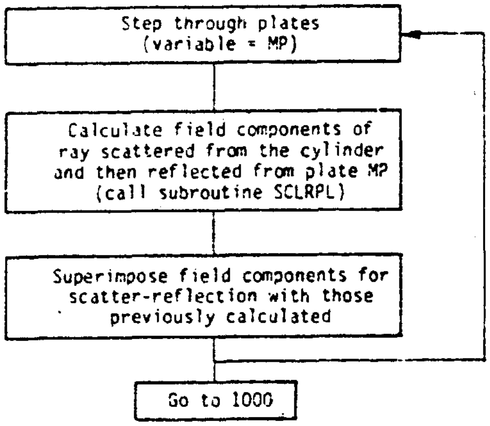
This section computes field components for all source rays which are reflected from a plate and then scattered by the cylinder.


Step through plates variable = MP
Is source a slot on the plate or a dipole mounted normal to the plate? (LSURF(MP)=TRUE)
Yes
No
Is plate MP totally shadowed from source?
Yes
No
Calculate field components for source ray which is reflected off of plate MP and then scattered by the cylinder (call subroutine RPLSCL)
Sum reflected scattered field components with those calculated previously
Go to 1000

```
250   CONTINUE
C!!!  COMPUTE ALL POSSIBLE FIELDS REFLECTED FROM THE PLATES THEN
C!!!  SCATTERED FROM THE CYLINDER
      DO 29 MP=1,MPXR
C!!!  IF ANTENNA IS ON PLATE, THEN NO REFLECTED FIELD
      IF(LSURF(MP)) GO TO 29
C!!!  IF PLATE SHADOWED, THEN NO REFLECTED FIELD
      IF(LSHD(MP)) GO TO 29
      CALL RPLSCL(ERPST,ERPSP,ERPCT,ERPCP,MP)
      ETH=ETH+ERPST
      EPH=EPH+ERPSP
      IF(.NOT.LOUT) GO TO 29
      CALL PRIOUT(240,MP,0,0,ERPCT,ERPCP)
      CALL PRIOUT(250,MP,0,0,ERPST,ERPSP)
29    CONTINUE
      GO TO 1000
```

K=3, J=2

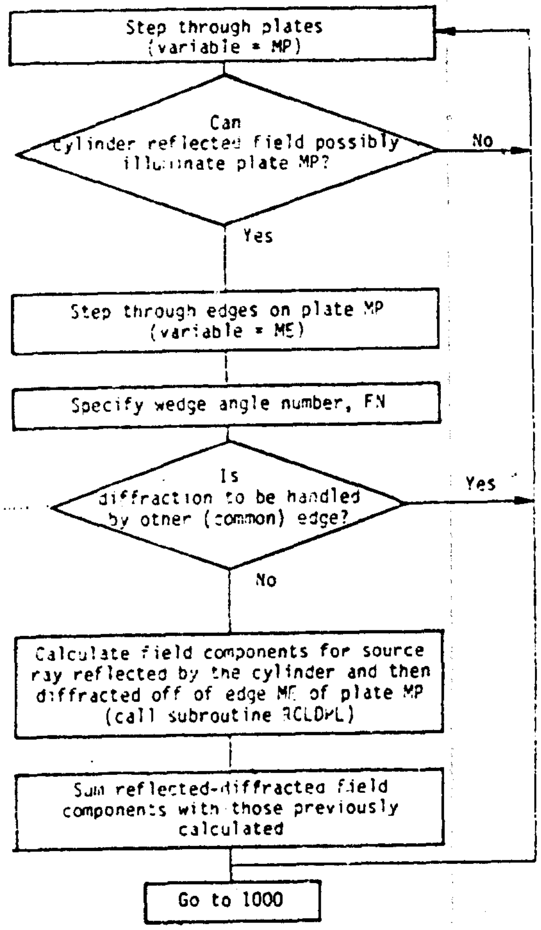
This section calculates field components for all source rays scattered from the cylinder and then reflected from a plate.


Step through plates
(variable = MP)
Calculate field components of ray scattered from the cylinder and then reflected from plate MP (call subroutine SCLRPL)
Superimpose field components for scatter-reflection with those previously calculated
Go to 1000

```
4ØØ    CONTINUE
C!!!   COMPUTE ALL POSSIBLE FIELDS SCATTERED FROM THE CYLINDER THEN
C!!!   REFLECTED FROM THE PLATES
       DO 4Ø MP=1,MPXR
       CALL SCLRPL(ERSPT,ERSPP,ERCPT,ERCPP,MP)
       ETH=ETH+ERSPT
       EPH=EPH+ERSPP
       IF(.NOT.LOUT) GO TO 4Ø
       CALL PRIOUT(41Ø,MP,Ø,Ø,ERCPT,ERCPP)
       CALL PRIOUT(42Ø,MP,Ø,Ø,ERSPT,ERSPP)
4Ø     CONTINUE
       GO TO 1ØØØ
```

K=3, J=3

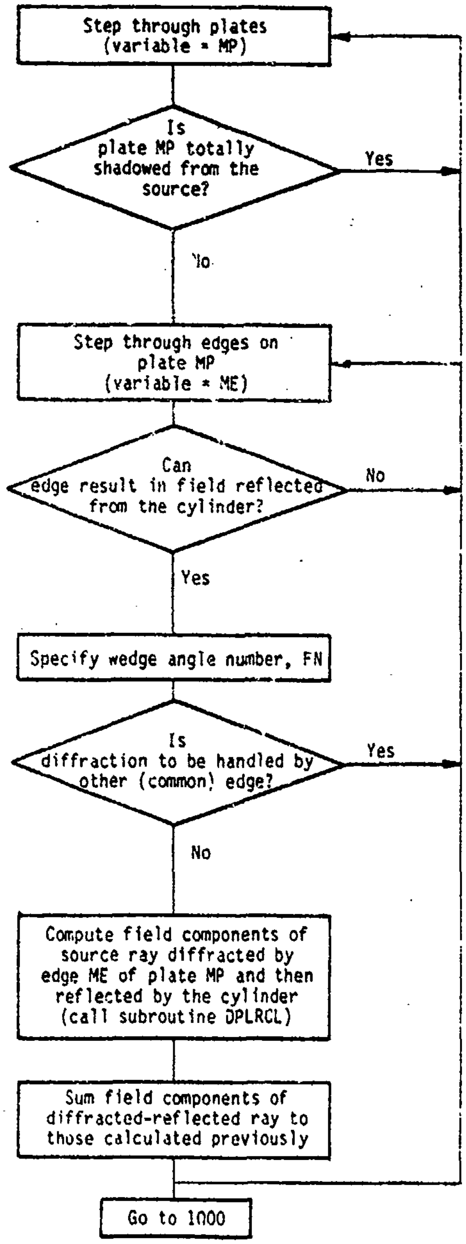
This section computes field components for all source rays reflected from the cylinder and diffracted from a plate edge.


Step through plates (variable = MP)
Can cylinder reflected field possibly illuminate plate MP?
No
Yes
Step through edges on plate MP (variable = ME)
Specify wedge angle number, FN
Is diffraction to be handled by other (common) edge?
Yes
No
Calculate field components for source ray reflected by the cylinder and then diffracted off of edge ME of plate MP (call subroutine RCLDPL)
Sum reflected-diffracted field components with those previously calculated
Go to 1000

```
 940   CONTINUE
C!!!   COMPUTE ALL POSSIBLE FIELDS REFLECTED FROM THE CYLINDER THEN
C!!!   DIFFRACTED FROM THE PLATES
       DO 91 MP=1,MPX
C!!!   IF PLATE SHADOWED, THEN NO DIFFRACTED FIELD
       IF(LSHD(MP)) GO TO 91
       MEX=MEP(MP)
       DO 90 ME=1,MEX
       FN=FNP(MP,ME)
       IF(FN.LT.0.) GO TO 90
       CALL RCLDPL(ERDTH,ERDPH,FN,ME,MP)
       ETH=ETH+ERDTH
       EPH=EPH+ERDPH
       IF(LOUT) CALL PRIOUT(940,MP,ME,0,ERDTH,ERDPH)
 90    CONTINUE
 91    CONTINUE
       GO TO 1000
```

K=3, J=4

This section computes field components of all source rays diffracted from plate edges and then reflected from the cylinder.


Step through plates (variable = MP)
Is plate MP totally shadowed from the source?
Yes
No
Step through edges on plate MP (variable = ME)
Can edge result in field reflected from the cylinder?
No
Yes
Specify wedge angle number, FN
Is diffraction to be handled by other (common) edge?
Yes
No
Compute field components of source ray diffracted by edge ME of plate MP and then reflected by the cylinder (call subroutine DPLRCL)
Sum field components of diffracted-reflected ray to those calculated previously
Go to 1000

```
950   CONTINUE
C!!!  COMPUTE ALL POSSIBLE FIELDS DIFFRACTED FROM THE PLATES THEN
C!!!  REFLECTED FROM THE CYLINDER
      DO 96 MP=1,MPX
C!!!  IF PLATE SHADOWED, THEN NO DIFFRACTED FIELD
      IF(LSHD(MP)) GO TO 96
      MEX=MEP(MP)
      DO 95 ME=1,MEX
C!!!  IF EDGE DOES NOT HAVE STRONG FIELD REFLECTED FROM CYLINDER
C!!!  BYPASS SUBR.
      IF(.NOT.LDC(MP,ME)) GO TO 95
      FN=FNP(MP,ME)
      IF(FN.LT.Ø.) GO TO 95
      CALL DPLRCL(EDRCT,EDRCP,FN,ME,MP)
      ETH=ETH+EDRCT
      EPH=EPH+EDRCP
      IF(LOUT) CALL PRIOUT(95Ø,MP,ME,Ø,EDRCT,EDRCP)
```

## SYMBOL DICTIONARY

| | |
|---|---|
| A | RADIUS OF CYLINDER ALONG X AXIS IN WAVELENGTHS |
| ALR | ANGLE THAT CONVERTS FIELD POLARIZATION FROM REFERENCE COORDINATE SYSTEM TO PATTERN CUT COORDINATE SYSTEM |
| AS | PI-THSR |
| AXCYL | VARIABLES USED TO DETERMINE IF THE CYLINDER COORDINATE SYSTEM IS THE SAME AS THE REFERENCE COORDINATE SYSTEM (BEFORE RCS TRANSFORMATION) |
| AYCYL | |
| AZCYL | |
| B | RADIUS OF CYLINDER ALONG Y AXIS IN WAVELENGTHS |
| BLR | BPL IN RADIANS |
| BPL | ANGLE THAT CONVERTS FIELD POLARIZATION FROM PATTERN CUT COORDINATE SYSTEM TO A RECEIVER COORDINATE SYSTEM (NOT PRESENTLY IMPLIMENTED) |
| CAS | COSINE OF AS |
| CNC | COSINE OF THTPR AND THTNR |
| CPS | COSINE OF PHSR |
| CTC | COTANGENT OF THTPR AND THTNR |
| CTHS | COSINE OF THSR |
| EDCPH | PHI COMPONENT OF FIELD DIFFRACTED FROM END CAP RIM IN RCS |
| EDCRPP | PHI COMPONENT OF FIELD DIFFRACTED FROM CORNERS OF EDGE ME OF PLATE MP AND THEN REFLECTED BY PLATE MR (CORNER DIF) |
| EDCTH | THETA COMPONENT OF FIELD DIFFRACTED FROM END CAP RIM IN RCS |
| EDCRPT | THETA COMPONENT OF FIELD DIFFRACTED FROM THE CORNERS OF EDGE ME OF PLATE MP AND THEN REFLECTED BY PLATE MR (CORNER DIFFRACTION) |
| EDPCPH | PHI COMPONENT OF FIELD DIFFRACTED FROM CORNERS OF EDGE ME OF PLATE MP |
| EDPCTH | THETA COMPONENT OF FIELD DIFFRACTED FROM CORNERS OF EDGE ME OF PLATE MP |
| EDPPH | PHI COMPONENT OF FIELD DIFFRACTED FROM EDGE ME OF PLATE MP IN RCS |
| EDPTH | THETA COMPONENT OF FIELD DIFFRACTED FROM EDGE ME OF PLATE MP IN RCS |
| EDRCP | PHI COMPONENT OF FIELD DIFFRACTED FROM EDGE ME OF PLATE MP AND REFLECTED FROM THE CYLINDER |
| EDRCT | THETA COMPONENT OF FIELD DIFFRACTED FROM EDGE ME OF PLATE MP AND REFLECTED FROM THE CYLINDER |
| EDRPP | PHI COMPONENT OF FIELD DIFFRACTED FROM EDGE ME OF PLATE ME AND THEN REFLECTED BY PLATE MR (EDGE DIF.) |
| EDRPT | THETA COMPONENT OF FIELD DIFFRACTED FROM EDGE ME OF PLATE MP AND THEN REFLECTED BY PLATE MR (EDGE DIFF.) |
| EIPH | PHI COMPONENT OF DIRECT FIELD FROM SOURCE IN RCS |
| EITH | THETA COMPONENT OF DIRECT FIELD FROM SOURCE IN RCS |
| EPH | PHI COMPONENT OF SCATTERED FIELD IN RCS |
| EPHT | PHI COMPONENT OF TOTAL CALCULATED E FIELD IN PATTERN CUT COORDINATE SYSTEM |
| ERCAP | PHI COMPONENT OF FIELD REFLECTED FROM CYLINDER END CAP IN RCS |
| ERCAT | THETA COMPONENT OF FIELD REFLECTED FROM CYLINDER END CAP IN RCS |
| ERCPP | PHI COMPONENT OF GEOMETRICAL OPTICS FIELD REFLECTED FROM CYLINDER, AND THEN REFLECTED FROM PLATE MR |
| ERCPT | THETA COMPONENT OF GEOMETRICAL OPTICS FIELD REFLECTED FROM CYLINDER, AND THEN REFLECTED FROM PLATE MR |
| ERDPH | PHI COMPONENT OF FIELD REFLECTED FROM CYLINDER AND DIFFRACTED BY EDGE ME OF PLATE MP |
| ERDTH | THETA COMPONENT OF FIELD REFLECTED FROM CYLINDER AND DIFFRACTED BY EDGE ME OF PLATE MP |
| ERPCP | PHI COMPONENT OF GEOMETRICAL OPTICS FIELD REFLECTED BY PLATE MR AND THEN SCATTERED BY THE CYLINDER |
| ERPCT | THETA COMPONENT OF GEOMETRICAL OPTICS FIELD REFLECTED BY PLATE MR AND THEN SCATTERED BY THE CYLINDER |
| ERPDCT | PHI COMPONENT OF FIELD REFLECTED BY PLATE MR AND |

| | |
|---|---|
| | DIFFRACTED BY THE CORNERS OF EDGE ME OF PLATE MP (CORNER DIFFRACTION) |
| ERPDCT | THETA COMPONENT OF FIELD REFLECTED BY PLATE MR AND DIFFRACTED BY THE CORNERS OF EDGE ME OF PLATE MP (CORNER DIFFRACTION) |
| ERPDP | PHI COMPONENT OF FIELD REFLECTED BY PLATE MR AND DIFFRACTED BY EDGE ME OF PLATE MP (EDGE DIFFRACTION) |
| ERPDT | THETA COMPONENT OF FIELD REFLECTED BY PLATE MR AND DIFFRACTED BY EDGE ME OF PLATE MP (EDGE DIFFRACTION) |
| ERPH | PHI COMPONENT OF GEOMETRICAL OPTICS FIELD REFLECTED FROM CYLINDER |
| ERPPH | PHI COMPONENT OF FIELD REFLECTED FROM PLATE MP IN RCS |
| ERPSP | PHI COMPONENT OF FIELD REFLECTED BY PLATE MR AND THEN SCATTERED BY THE CYLINDER |
| ERPST | THETA COMPONENT OF FIELD PEFLECTED BY PLATE MR, AND THEN SCATTERED BY THE CYLINDER |
| ERPTH | THETA COMPONENT OF FIELD REFLECTED FROM PLATE MP |
| ERRPP | PHI COMPONENT OF FIELD REFLECTED FROM PLATE MP AND THEN PLATE MPP IN RCS |
| ERRPT | THETA COMPONENT OF FIELD REFLECTED FROM PLATE MP AND THEN PLATE MPP IN RCS |
| ERSPP | PHI COMPONENT OF FIELD SCATTERED BY THE CYLINDER AND THEN REFLECTED BY PLATE MR |
| ERSPT | THETA COMPONENT OF FIELD SCATTERED BY THE CYLINDER AND THEN REFLECTED BY PLATE MR |
| ERTH | THETA COMPONENT OF GEOMETRICAL OPTICS FIELD REFLECTED FROM CYLINDER |
| ESPH | PHI COMPONENT OF FIELD SCATTERED BY CYLINDER IN RCS |
| ESTH | THETA COMPONENT OF FIELD SCATTERED BY CYLINDER IN RCS |
| ETH | THETA COMPONENT OF SCATTERED FIELD IN RCS |
| ETHT | THETA COMPONENT OF TOTAL CALCULATED E FIELD IN PATTERN CUT COORDINATE SYSTEM |
| FN | WEDGE ANGLE INDICATOR OF EDGE ME OF PLATE MP |
| FRQG | THE FREQUENCY IN GIGAHERTZ |
| I | DO LOOP VARIABLE |
| IBP | PATTERN ANGLE LOWER LIMIT PLUS ONE |
| IEP | PATTERN ANGLE UPPER LIMIT PLUS ONE |
| II | DO LOOP VARIABLE USED TO STEP THROUGH PATTERN ANGLE |
| IR | CHARACTER STRING USED TO INPUT COMMAND DESIRED |
| IS | INCREMENT ON PATTERN ANGLE |
| IT | CHARACTER STRINGS CONTAINING COMMAND VARIABLES FOR DATA INPUT |
| ITT | CHARACTER STRINGS USED AS COMMAND VARIABLES FOR DATA INPUT |
| J | DO LOOP VARIABLE USED TO STEP THROUGH INDIVIDUAL GTD TERMS |
| K | DO LOOP VARIABLE USED TO STEP THRU MAJOR GTD GROUPINGS |
| LABEL | CHARACTERS USED TO SPECIFY UNITS USED TO INPUT DATA |
| LAMP | LOGICAL VARIABLE SET TRUE IF NEC SOURCE DATA WAS INPUT |
| LNROT | LOGICAL VARIABLE: SET TRUE IF RCS TRANSFORMATION IS NOT TO BE PERFORMED |
| MC | INDEX VARIABLE FOR CORNERS |
| ME | INDEX VARIABLE FOR EDGES |
| MEDX | MAXIMUM NUMBER OF EDGES ALLOWED ON ONE PLATE |
| MEX | NUMBER OF EDGES ON PLATE MP (NOT AN ARRAY) |
| MP | INDEX VARIABLE FOR PLATES |
| MPDX | MAXIMUM NUMLER OF PLATES ALLOWED |
| MPP | INDEX VARIABLE FOR PLATES |
| MR | INDEX VARIABLE FOR PLATES |
| MS | INDEX VARIABLE FOR SOURCES |
| MSDX | MAXIMUM NUMBER OF SOURCES ALLOWED |
| N | INDEX VARIABLE |
| NI | INDEX VARIABLE |
| NJ | INDEX VARIABLE |
| PHP | PHI ANGLE DEFINING PATTERN ANGLE IN PATTERN CUT |

| | |
|---|---|
| | COORDINATE SYSTEM |
| PHPR | PHI COMPONENT OF PATTERN ANGLE IN PAT CUT COORD SYS |
| PHSR | PHI COMPONENT OF PATTERN (OBSERVATION) ANGLE IN RCS |
| SAS | SINE OF AS |
| SASP | SIN(AS-PI/2) |
| SNC | SINE OF THTPR AND THTNR |
| SPS | SINE OF PHSR |
| STHS | SINE OF THSR |
| THP | THETA ANGLE DEFINING PATTERN ANGLE IN PATTERN CUT COORDINATE SYSTEM |
| THPR | THETA COMPONENT OF PATTERN ANGLE IN PAT CUT COORD SYS |
| THSR | THETA COMPONENT OF PATTERN (OBSERVATION) ANGLE IN RCS |
| THTNR | ANGLE NEGATIVE END CAP MAKES WITH Z AXIS (IN X-Z PLANE) |
| THTPR | ANGLE POSITIVE END CAP MAKES WITH Z AXIS (IN X-Z PLANE) |
| TPPD | PATTERN ANGLE WHICH REMAINS CONSTANT |
| UNIT | CONVERSION FACTORS TO CONVERT FROM METERS, FEET, OR INCHES TO METERS |
| VXS | X,Y,Z COMPONENTS DEFINING SOURCE COORDINATE AXES IN RCS COMPONENTS |
| WI | (COMPLEX) WEIGHTING COEFFICIENT OF SOURCE EXCITATION |
| XCL | X,Y,Z COMPONENTS DEFINING AXES OF CYLINDER COORDINATE SYSTEM (BEFORE RCS TRANSFORMATION) (IN RCS COMPONENTS) |
| YCL | |
| ZCL | |
| XCC | X,Y,Z COMPONENTS OF LOCATION OF CYLINDER COORDINATE SYSTEM ORIGIN IN RCS (BEFORE RCS TRANSFORMATION) |
| XCOM | DISTANCE BETWEEN RCS ORIGIN AND CYLINDER COORDINATE SYSTEM ORIGIN |
| XOO | CONSTANT (=Ø,Ø,Ø) |
| XPC | X,Y,Z COMPONENTS DEFINING AXES OF PATTERN CUT COORDINATE SYSTEM AFTER RCS TRANSFORMATION (IN RCS COMPONENTS) |
| YPC | |
| ZPC | |
| XPD | X,Y,Z COMPONENTS DEFINING AXES OF PATTERN CUT COORDINATE SYSTEM (IN RCS COMPONENTS) (BEFORE RCS TRANSFORMATION) |
| YPD | |
| ZPD | |
| XS | X,Y,Z COMPONENTS OF SOURCE LOCATION (INSIDE SOURCE LOOP) |
| XXX | COMPUTATIONAL VARIBLE |
| ZC | POINT WHERE UPPER AND LOWER CYLINDER END CAPS MEET THE Z AXIS OF THE RCS |

## BABS

### PURPOSE

This function computes the absolute value of a complex argument. It is similar to CABS, except it avoids run time errors when the real part and imaginary part of the argument are zero.

### METHOD

The system function CABS is used unless the absolute value of the real part and the imaginary part of the argument are close to zero, in which case a very small value is returned.

### SYMBOL DICTIONARY

| | |
|---|---|
| X | ABSOLUTE VALUE OF THE REAL PART OF Z |
| Y | ABSOLUTE VALUE OF THE IMAGINARY PART OF Z |
| Z | THE COMPLEX ARGUMENT |

### CODE LISTING

```
C-----------------------------------------------------------------
      FUNCTION BABS(Z)
C!!!
C!!!  THIS ROUTINE IS USED TO GIVE COMPLEX ABSOLUTE VALUES. IT IS
C!!!  USED RATHER THAN STANDARD ROUTINES TO AVOID EXECUTION
C!!!  ERRORS.
C!!!
      COMPLEX Z
      X=ABS(REAL(Z))
      Y=ABS(AIMAG(Z))
      IF(X.LT.1.E-10.AND.Y.LT.1.E-10) GO TO 10
      BABS=CABS(Z)
      RETURN
 10   BABS=1.E-10
      RETURN
      END
```

## BLOCK DATA

### PURPOSE

To load commonly used data into the common area.

### CODE LISTING

```
C-----------------------------------------------------------------
      BLOCK DATA
C!!!
C!!!  LOAD COMMONLY USED DATA INTO COMMON AREA.
C!!!
      COMPLEX CJ,CPI4,TOP
      COMMON/PIS/PI,TPI,DPR,RPD
      COMMON/COMP/CJ,CPI4
      COMMON/TOPD/TOP
      DATA PI,TPI,DPR,RPD/3.14159265,6.28318531,57.2957795,
     20.0174532925/
      DATA CJ,CPI4/(0.,1.),(.70710678,-.70710678)/
      DATA TOP/(-.70710678,.70710678)/
      END
```

## BLOG10

PURPOSE

This function computes the logarithm to the base ten of the argument. It is similar to ALOG10, except it avoids run time errors when the argument is zero.

METHOD

The system function ALOG10 is used unless the argument is close to zero, in which case the logarithm of the limit number is returned.

SYMBOL DICTIONARY

```
X      THE ARGUMENT OF THE FUNCTION
```

CODE LISTING

```
C-------------------------------------------------------------
      FUNCTION BLOG10(X)
C!!!
C!!!  THIS ROUTINE AVOIDS THE ERROR ASSOCIATED WITH THE
C!!!  ALOG10 OF A ZERO NUMBER.
C!!!
      IF(X.GT.1.E-10) GO TO 1
      BLOG10=-10.
      RETURN
    1 BLOG10=ALOG10(X)
      RETURN
      END
```

## BTAN2

### PURPOSE

This function computes the two argument arctangent function. It is similar to ATAN2, except it avoids run time errors when the second argument is zero.

### METHOD

The system function ATAN2(Y,X) is used to return the angle in radians, whose sine is Y and cosine is X unless the second argument or both of the arguments are zero. If the second argument is zero, either π/2 or -π/2 is returned depending on the sign of the first argument. If both arguments are zero, a zero value is returned.

### SYMBOL DICTIONARY

X SECCND ARGUMENT, WHICH IS THE COSINE OF THE ANGLE TO BE COMPUTED

Y FIRST ARGUMENT, WHICH IS THE SINE OF THE ANGLE TO BE COMPUTED

### CODE LISTING

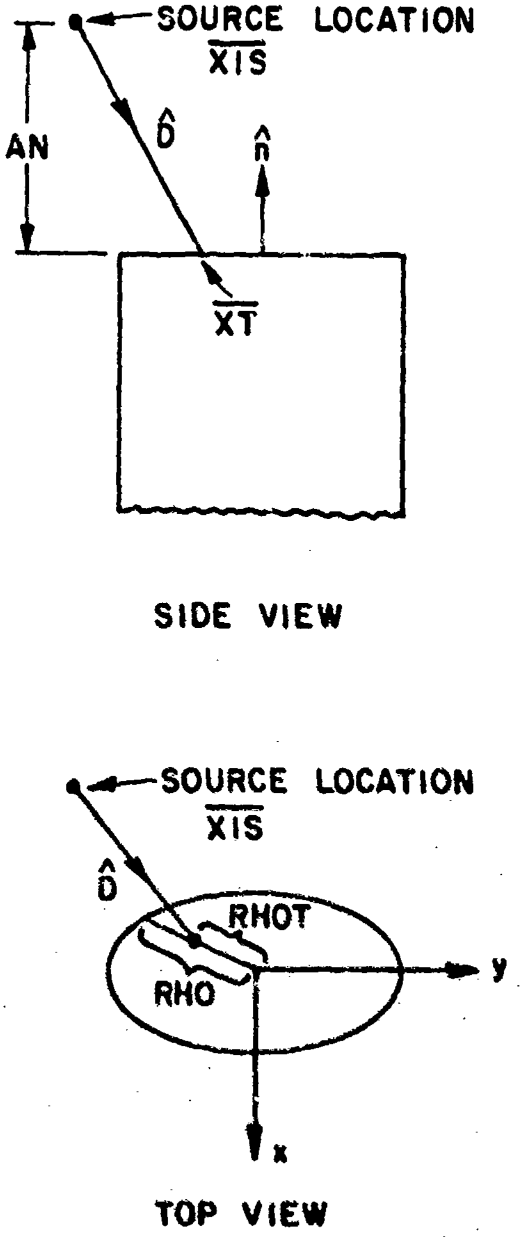
```
C-----------------------------------------------------------
      FUNCTION BTAN2(Y,X)
C!!!
C!!!  THIS ROUTINE IS USED TO COMPUTE THE ARCTANGENT. IT IS
C!!!  SIMILAR TO ATAN2 EXCEPT IT AVOIDS THE RUN TIME ERRORS.
C!!!
      COMMON/PIS/PI,TPI,DPR,RPD
      IF(ABS(X).GT.1.E-10) GO TO 50
      IF(ABS(Y).GT.1.E-10) GO TO 10
      BTAN2=0.
      RETURN
 10   BTAN2=PI/2.
      IF(Y.LT.0.) BTAN2=-BTAN2
      RETURN
 50   BTAN2=ATAN2(Y,X)
      RETURN
      END
```

## CAPINT

### PURPOSE

To determine if a ray traveling from a given source location in a given direction will hit a cylinder end cap.

### PERTINENT GEOMETRY



SIDE VIEW

TOP VIEW

Figure 48--Geometry of ray which hits an end cap.

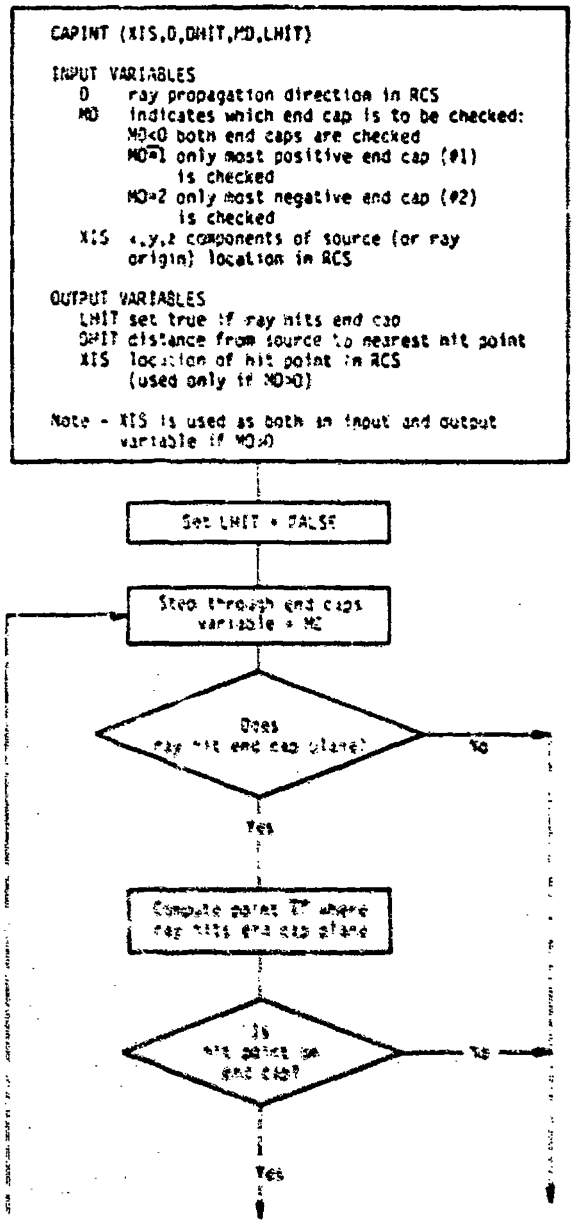
## METHOD

The subroutine checks to see if a ray emanating from a source in a given direction hits a cylinder end cap. First it checks if the ray is aimed toward or away from the end cap plane by comparing the sign of the dot product of the scatter direction and end cap normal (DN) and the sign of the dot product of the source location vector and end cap normal (AN). If the ray is directed toward the end cap plane as shown in Figure 48, the intersetion point with the plane is found from

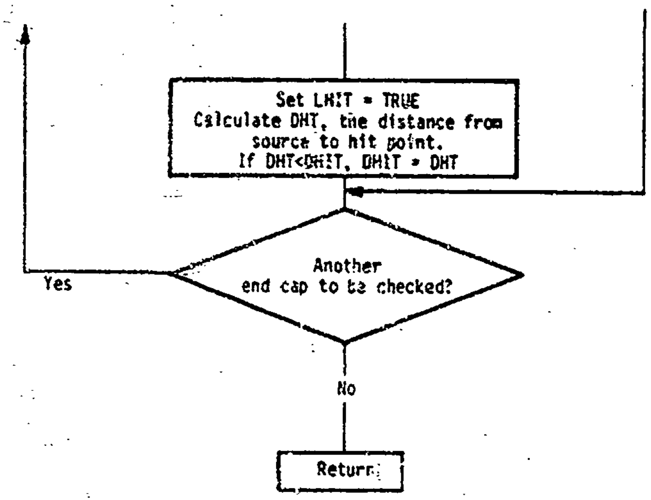
$$\overline{XT} = \overline{XIS} - \hat{D}\,\frac{AN}{DN}\ .$$

The distance from the intersection point to the center of the end cap is then compared with the radius of the end cap to determine if the intersection point lies within the finite limits of the end cap.

FLOW DIAGRAM



```
CAPINT (XIS,D,DHIT,MD,LHIT)

INPUT VARIABLES
   D    ray propagation direction in RCS
   MD   indicates which end cap is to be checked:
        MD<0 both end caps are checked
        MD=1 only most positive end cap (#1)
             is checked
        MD=2 only most negative end cap (#2)
             is checked
   XIS  x,y,z components of source (or ray
        origin) location in RCS

OUTPUT VARIABLES
   LHIT set true if ray hits end cap
   DHIT distance from source to nearest hit point
   XIS  location of hit point in RCS
        (used only if MD>0)

Note - XIS is used as both an input and output
       variable if MD>0
```


Set LHIT = TRUE
Calculate DHT, the distance from source to hit point.
If DHT<DHIT, DHIT = DHT
Another end cap to be checked?
Yes
No
Return

SYMBOL DICTIONARY

| | |
|---|---|
| AE | DISTANCE FROM CENTER OF EDGE CAP TO EDGE ALONG LINE IN X-Z PLANE |
| AN | DOT PRODUCT OF VECTOR FROM END CAP TO SOURCE AND END CAP UNIT NORMAL |
| CVE | COSINE OF VE |
| D | PROPAGATION DIRECTION IN RCS |
| DHIT | DISTANCE FROM SOURCE TO NEAREST HIT POINT |
| DHT | DISTANCE FROM SOURCE TO HIT POINT |
| DN | DOT PRODUCT OF END CAP UNIT NORMAL AND THE RAY PROPAGATION DIRECTION |
| LHIT | SET TRUE IF RAY HITS END CAP |
| MC | END CAP INDEX VARIABLE |
| MD | INDICATES WHICH END CAPS ARE TO BE CHECKED |
| NC | SIGN CHANGE VARIABLE |
| RHO | DISTANCE FROM Z AXIS TO POINT WHERE RAY CONNECTING THE HIT POINT AND THE ORIGIN HITS THE CYLINDER (2-D) |
| RHOT | DISTANCE FROM Z AXIS TO POINT XT |
| SVE | SINE OF VE |
| VE | ELL ANGLE DEFINING HIT POINT |
| XIS | (ENTERING ROUTINE) SOURCE LOCATION<br>(LEAVING ROUTINE) HIT POINT (IF MD>0) |
| XT | X,Y,Z COMPONENTS OF POINT WHERE RAY HITS END CAP PLANE |

## CODE LISTING

```
C-----------------------------------------------------------------------
      SUBROUTINE CAPINT(XIS,D,DHIT,MD,LHIT)
C!!!
C!!!  DOES RAY HIT END CAP?
C!!!
      DIMENSION XIS(3),D(3),XT(3)
      LOGICAL LHIT,LDEBUG,LTEST
      COMMON/GEOMEL/A,B,ZC(2),SNC(2),CNC(2),CTC(2)
      COMMON/TEST/LDEBUG,LTEST
      LHIT=.FALSE.
      DHIT=Ø.
C!!!  STEP THRU END CAPS
      DO 4Ø MC=1,2
      IF(MD.NE.Ø.AND.MC.NE.MD) GO TO 4Ø
      NC=MC
      IF(MC.EQ.2) NC=-1
      AN=-XIS(1)*NC*CNC(MC)+(XIS(3)-ZC(MC))*NC*SNC(MC)
      DN=-NC*CNC(MC)*D(1)+NC*SNC(MC)*D(3)
C!!!  DOES RAY HIT END CAP PLANE?
      IF(AN*DN.GE.Ø.) GO TO 4Ø
C!!!  COMPUTE POINT XT, WHERE RAY HITS END CAP PLANE
      DO 1Ø N=1,3
1Ø    XT(N)=XIS(N)-AN*D(N)/DN
      RHOT=XT(1)*XT(1)+XT(2)*XT(2)+(XT(3)-ZC(MC))*(XT(3)-ZC(MC))
      RHOT=SQRT(RHOT)
      AE=A/SNC(MC)
C!!!  IS HIT POINT ON END CAP?
      IF(RHOT,GT.AE.AND.RHOT.GT.B) GO TO 4Ø
      IF(RHOT.LT.AE.AND.RHOT.LT.B) GO TO 2Ø
      VE=BTAN2(A*XT(2),B*XT(1))
      CVE=COS(VE)
      SVE=SIN(VE)
      RHO=SQRT(AE*AE*CVE*CVE+B*B*SVE*SVE)
      IF(RHOT.GT.RHO) GO TO 4Ø
2Ø    CONTINUE
C!!!  CALCULATE DHT, THE DISTANCE FROM SOURCE TO HIT POINT
      DHT=Ø.
      DO 3Ø N=1,3
3Ø    DHT=DHT+(XT(N)-XIS(N))*(XT(N)-XIS(N))
      DHT=SQRT(DHT)+1.E-5
      IF(LHIT.AND.(DHT.GT.DHIT)) GO TO 4Ø
      LHIT=.TRUE.
      DHIT=DHT
      IF(MD.LE.Ø) GO TO 4Ø
      DO 35 N=1,3
35    XIS(N)=XT(N)
4Ø    CONTINUE
      IF(.NOT.LTEST) RETURN
      WRITE(6,9ØØ)
9ØØ   FORMAT(/,' TESTING CAPINT SUBROUTINE')
      WRITE(6,*) XIS
      WRITE(6,*) D
      WRITE(6,*) DHIT,MD,LHIT
      RETURN
      END
```

## CYLINT

### PURPOSE

To determine if a ray travelling from a given source location in a given direction will intersect the elliptic cylinder.

### PERTINENT GEOMETRY

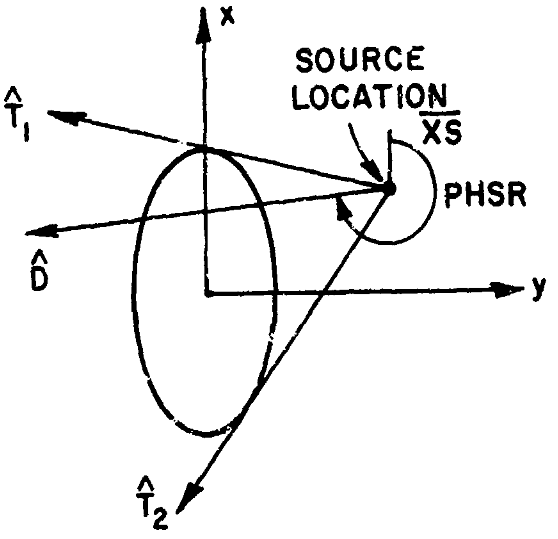


Figure 49a--Illustration of ray that hits infinite cylinder.

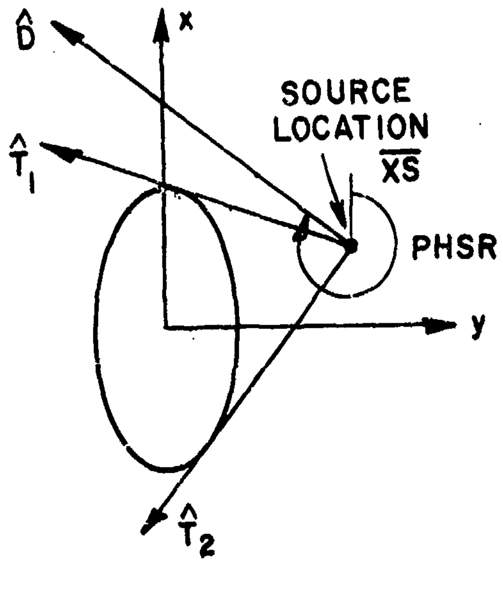


Figure 49b--Illustration of ray that doesn't hit finite cylinder.

$\hat{T}_1 = \hat{x}\ BT(1) + \hat{y}\ BT(2)$

$\hat{T}_2 = \hat{x}\ BT(3) + \hat{y}\ BT(4)$

$\hat{D}\ = \hat{x}\ D(1) + \hat{y}\ D(2) + \hat{z}\ D(3)$

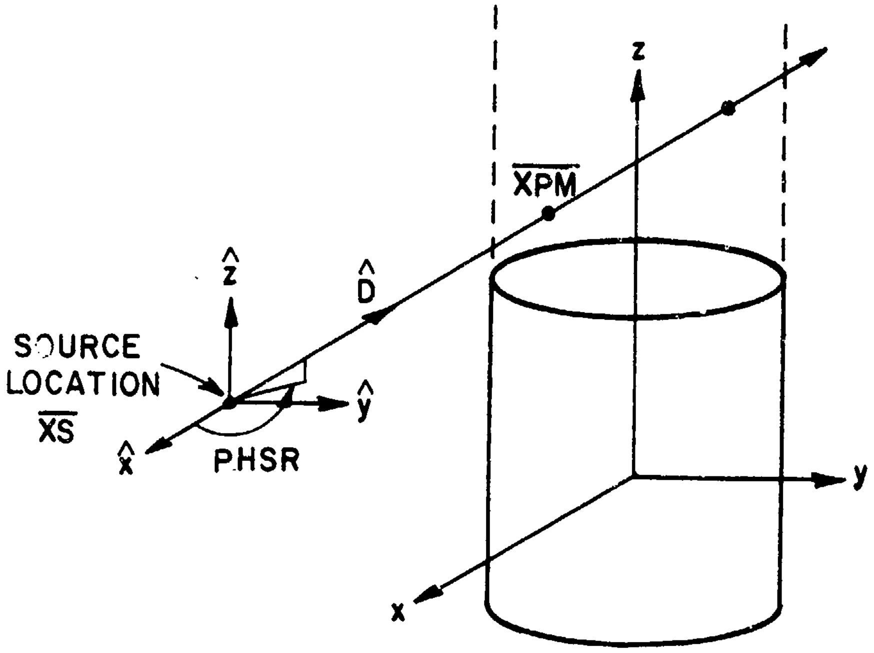


Figure 50a--Illustration of ray that hits infinite cylinder but not finite cylinder.

$\overline{XPM} = \hat{x}\ XPM(1) + \hat{y}\ XPM(2) + \hat{z}\ XPM(3)$

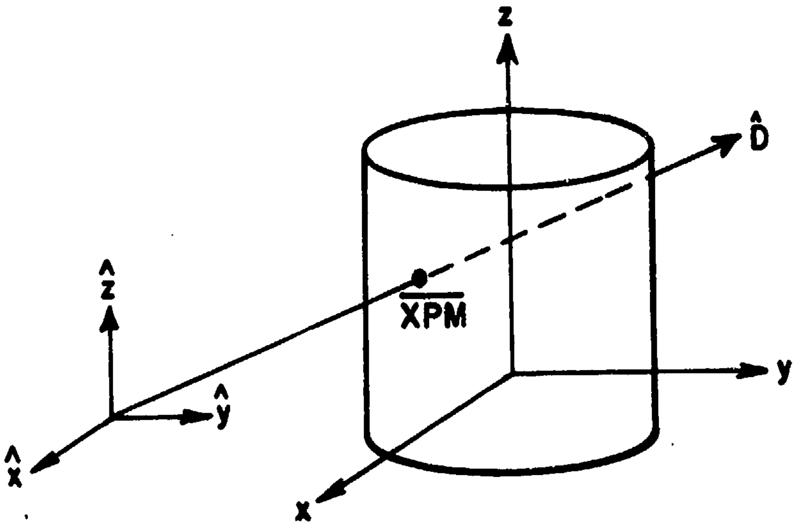


Figure 50b--Illustration of ray that hits finite cylinder.

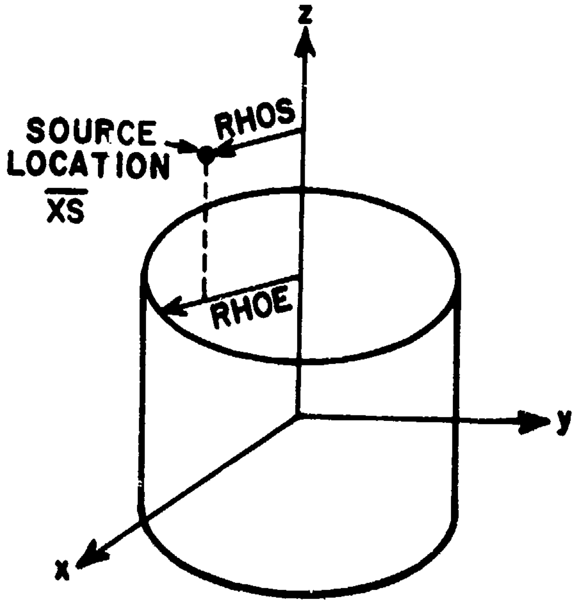


Figure 51--Illustration of source which cannot illuminate curved cylinder surface. RHOS<RHOE.

METHOD

This subroutine determines if a ray eminating from a source in a given direction hits the finite elliptic cylinder. First the distance from the source to the cylinder axis is compared to the radius of the cylinder to see if the source can illuminate the curved surface of the cylinder as illustrated in Figure 51. If it can not, then the subroutine checks whether the ray hits an end cap. If it is possible to hit the curved surface, the ray is checked to see whether or not it is aimed in the direction of the infinite cylinder as shown in Figure 49.

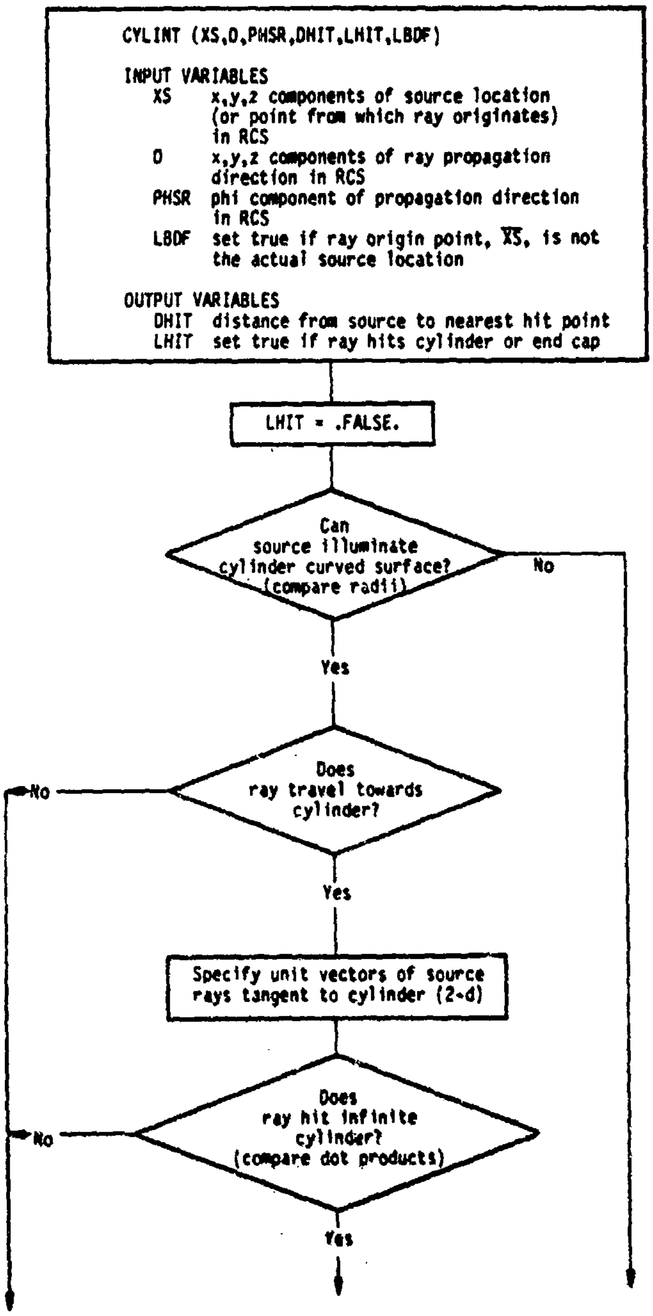
If the ray travels towards the cylinder, the routine compares dot products in order to determine if the ray will hit the infinite cylinder:

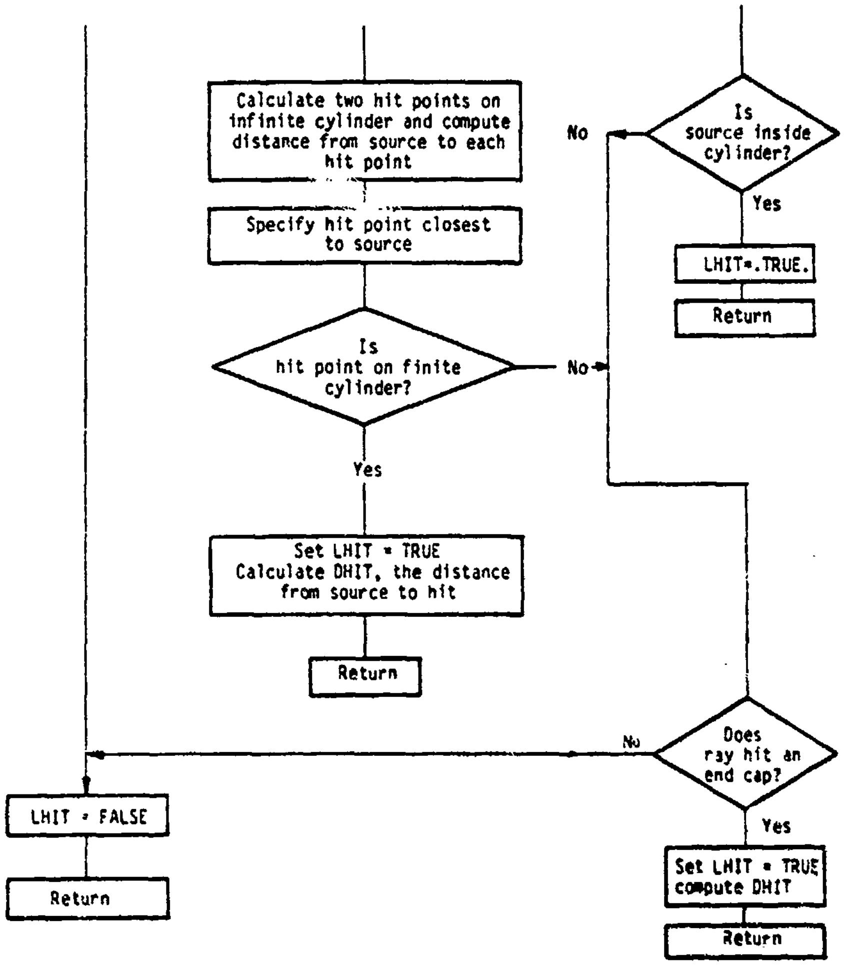
If $\hat{D} \cdot \hat{T}_1 \geq \hat{T}_1 \cdot \hat{T}_2$ and $\hat{D} \cdot \hat{T}_2 \geq \hat{T}_1 \cdot \hat{T}_2$, the ray hits the infinite cylinder (see Figure 49a).

If $\hat{D} \cdot \hat{T}_1 < \hat{T}_1 \cdot \hat{T}_2$ or $\hat{D} \cdot \hat{T}_2 < \hat{T}_1 \cdot \hat{T}_2$, the ray does not hit the infinite cylinder (see Figure 49b).

The subroutine then solves a quadratic equation to determine the intersection point. The details are given on pages 90-96 of Reference 1. A test is then made as to whether or not this intersection point lies on or off the limits of the finite cylinder (see Figures 50a and 50b).

## FLOW DIAGRAM


CYLINT (XS,D,PHSR,DHIT,LHIT,LBDF)
INPUT VARIABLES
XS x,y,z components of source location (or point from which ray originates) in RCS
D x,y,z components of ray propagation direction in RCS
PHSR phi component of propagation direction in RCS
LBDF set true if ray origin point, $\overline{XS}$, is not the actual source location
OUTPUT VARIABLES
DHIT distance from source to nearest hit point
LHIT set true if ray hits cylinder or end cap
LHIT = .FALSE.
Can source illuminate cylinder curved surface? (compare radii)
No
Yes
Does ray travel towards cylinder?
No
Yes
Specify unit vectors of source rays tangent to cylinder (2-d)
Does ray hit infinite cylinder? (compare dot products)
No
Yes


Calculate two hit points on infinite cylinder and compute distance from source to each hit point
Specify hit point closest to source
Is hit point on finite cylinder?
No
Yes
Set LHIT = TRUE Calculate DHIT, the distance from source to hit
Return
Is source inside cylinder?
No
Yes
LHIT=.TRUE.
Return
Does ray hit an end cap?
No
Yes
Set LHIT = TRUE compute DHIT
Return
LHIT = FALSE
Return

## SYMBOL DICTIONARY

| | |
|---|---|
| BM | PARAMETER USED IN COMPUTING HIT POINT 1 |
| BPL | PARAMETER USED IN COMPUTING HIT POINT 2 |
| BTD | X AND Y COMPONENTS OF UNIT VECTORS OF SOURCE RAYS TANGENT TO CYLINDER |
| CPS | COSINE OF PHSR |
| CVE | COSINE OF VE |
| D | RAY PROPAGATION DIRECTION IN REF COORD SYS |
| D12 | DOT PRODUCT OF SOURCE VECTORS TANGENT TO THE CYLINDER (IN X-Y PLANE) |
| DD1 | DOT PRODUCT OF THE PROPAGATION DIRECTION AND T1 TANGENT UNIT VECTOR |
| DD2 | DOT PRODUCT OF THE PROPAGATION DIRECTION AND T2 TANGENT UNIT VECTOR |
| DHIT | DISTANCE FROM SOURCE TO (NEAREST) HIT POINT |
| DM | DISTANCE FROM SOURCE TO HIT POINT 1 |
| DPL | DISTANCE FROM SOURCE TO HIT POINT 2 |
| DTD | DOT PRODUCT OF SOURCE VECTORS TANGENT TO THE CYLINDER (X-Y PLANE) |
| DXY | DOT PRODUCT OF RAY FROM ORIGIN TO SOURCE AND PROPAGATION DIRECTION (IN X-Y PLANE) |
| F | COMPUTATIONAL VARIABLE |
| FG | COMPUTATIONAL VARIABLE |
| FGH | COMPUTATIONAL VARIABLE |
| FH | COMPUTATIONAL VARIABLE |
| G | COMPUTATIONAL VARIABLE |
| GH | COMPUTATIONAL VARIABLE |
| H | COMPUTATIONAL VARIABLE |
| LBDF | SET TRUE IF RAY ORIGIN XS IS NOT THE SOURCE LOCATION |
| LHIT | SET TRUE IF RAY HITS CYLINDER OR END CAP |
| PHSR | PHI COMPONENT OF PROPAGATION DIRECTION IN RCS |
| RHOE | RADIUS FROM Z AXIS TO POINT WHERE RAY FROM ORIGIN TO SOURCE INTERSECTS THE CYLINDER |
| RHOS | DISTANCE FROM SOURCE TO Z AXIS |
| SPS | SINE OF PHSR |
| SVE | SINE OF VE |
| TOP | COMPUTATIONAL VARIABLE |
| TX1 | X COMPONTNT OF TANGENT UNIT VECTOR, T1 |
| TX2 | X COMPONENT OF TANGENT UNIT VECTOR, T2 |
| TY1 | Y COMPONENT OF TANGENT UNIT VECTOR, T1 |
| TY2 | Y COMPONENT OF TANGENT UNIT VECTOR, T2 |
| VE | ELL ANGLE OF SOURCE LOCATION IN ERCS |
| VM | ELL ANGLE DEFINING HIT POINT 1 ON CYLINDER IN ERCS |
| VPL | ELL ANGLE DEFINING HIT POINT 2 ON CYLINDER IN ERCS |
| VT | ELL ANGLE DEFINING HIT POINT ON CYLINDER CLOSEST TO SOURCE |
| VTD | NOT USED |
| XPM | USED IN SEVERAL CASES TO DEFINE HIT POINT |
| YPM | (X,Y,Z COMPONENTS IN RCS) ON CYLINDER |
| ZPM | (USED IN VARIOUS FORMS) |
| XS | SOURCE LOCATION (OR POINT FROM WHICH RAY ORIGINATES) IN RCS |

## CODE LISTING

```
C------------------------------------------------------------------------
      SUBROUTINE CYLINT(XS,D,PHSR,DHIT,LHIT,LBDF)
C!!!
C!!!  DOES RAY HIT CYLINDER?
C!!!
      DIMENSION D(3),XS(3),VTD(2),BTD(4)
      LOGICAL LHIT,LBDF,LPLA,LCYL,LDEBUG,LTEST
      COMMON/GEOMEL/A,B,ZC(2),SNC(2),CNC(2),CTC(2)
      COMMON/PIS/PI,TPI,DPR,RPD
      COMMON/ENDSCL/DTS,VTS(2),BTS(4)
      COMMON/LPLCY/LPLA,LCYL
      COMMON/TEST/LDEBUG,LTEST
      LHIT=.FALSE.
      DHIT=0.
      IF(.NOT.LCYL) GO TO 50
      RHOS=SQRT(XS(1)*XS(1)+XS(2)*XS(2))
C!!!  CAN SOURCE ILLUMINATE CYLINDER SURFACE?
      IF(RHOS.GT.A.AND.RHOS.GT.B) GO TO 5
      IF(RHOS.LT.A.AND.RHOS.LT.B) GO TO 30
      VE=ATAN2(A*XS(2),B*XS(1))
      CVE=COS(VE)
      SVE=SIN(VE)
      RHOE=SQRT(A*A*CVE*CVE+B*B*SVE*SVE)
      IF(RHOS.LE.RHOE) GO TO 30
    5 CONTINUE
      CPS=COS(PHSR)
      SPS=SIN(PHSR)
      DXY=XS(1)*CPS+XS(2)*SPS
C!!!  DOES RAY TRAVEL TOWARDS CYLINDER?
C!!!  (CHECK SIGN OF DOT PRODUCT OF PROP. DIR AND
C!!!  SOURCE LOCATION VECTOR)
      IF(DXY.GT.0.) GO TO 50
      IF(LBDF) GO TO 10
C!!!  SPECIFY CYLINDER TANGENT UNIT VECTORS
      D12=DTS
      TX1=BTS(1)
      TY1=BTS(2)
      TX2=BTS(3)
      TY2=BTS(4)
      GO TO 20
   10 CALL TANG(DTD,VTD,BTD,XS)
      D12=DTD
      TX1=BTD(1)
      TY1=BTD(2)
      TX2=BTD(3)
      TY2=BTD(4)
   20 CONTINUE
      DD1=CPS*TX1+SPS*TY1
      DD2=CPS*TX2+SPS*TY2
C!!!  COMPARE DOT PRODUCTS TO DETERMINE IF RAY HITS
C!!!  INFINITE CYLINDER
      IF(DD1.LT.D12.OR.DD2.LT.D12) GO TO 50
      F=A*SPS
      G=-B*CPS
      H=XS(1)*SPS-XS(2)*CPS
      FH=F*H
      FG=F*F+G*G
      GH=G*G-H*H
      FGH=FH*FH+FG*GH
      IF(FGH.LT.0.) GO TO 50
      BPL=((FH+SQRT(FGH))/FG)
      BM=((FH-SQRT(FGH))/FG)
      TOP=(-F*BPL+H)/G
      VPL=ATAN2(TOP,BPL)
      TOM=(-F*BM+H)/G
C!!!  COMPUTE TWO HIT POINTS AND COMPUTE DISTANCE
```

```
C!!!  FROM SOURCE TO EACH POINT
      VM=BTAN2(TOP,BM)
      XPM=A*COS(VPL)
      YPM=B*SIN(VPL)
      DPL=SQRT((XPM-XS(1))**2+(YPM-XS(2))**2)
      XPM=A*COS(VM)
      YPM=B*SIN(VM)
      DM=SQRT((XPM-XS(1))**2+(YPM-XS(2))**2)
C!!!  SPECIFY HIT POINT CLOSEST TO SOURCE
      VT=VM
      IF(DPL.LE.DM) VT=VPL
      XPM=A*COS(VT)
      ZPM=D(3)*(XPM-XS(1))/D(1)
      ZPS=ZPM+XS(3)
C!!!  IS HIT POINT ON FINITE CYLINDER?
      IF(ZPS.GT.ZC(1)+XPM*CTC(1).OR.
     2ZPS.LT.ZC(2)+XPM*CTC(2)) GO TO 40
      XPM=XPM-XS(1)
      YPM=B*SIN(VT)-XS(2)
C!!!  CALCULATE DISTANCE FROM SOURCE TO HIT
      DHIT=SQRT(XPM*XPM+YPM*YPM+ZPM*ZPM)+1.E-5
      LHIT=.TRUE.
      GO TO 50
   30 CONTINUE
C!!!  IF SOURCE CANNOT ILLUMINATE CYLINDER SIDES, IS SOURCE
C!!!  INSIDE CYLINDER?
      IF(XS(3).GT.(ZC(1)+XS(1)*CTC(1))) GO TO 40
      IF(XS(3).LT.(ZC(2)+XS(1)*CTC(2))) GO TO 40
      LHIT=.TRUE.
      GO TO 50
   40 CONTINUE
C!!!  IF RAY IS NOT SHADOWED BY CYLINDER, CHECK TO SEE IF RAY
C!!!  HITS END CAP
      CALL CAPINT(XS,D,DHIT,0,LHIT)
   50 IF(.NOT.LTEST) RETURN
      WRITE(6,900)
  900 FORMAT(/,' TESTING CYLINT SUBROUTINE')
      WRITE(6,*) XS
      WRITE(6,*) D
      WRITE(6,*) PHSR,DHIT,LHIT,LDDF
      RETURN
      END
```

DFPTCL

PURPOSE

To determine the four diffraction points which can occur on a cylinder end cap rim for a given radiation direction $\hat{D}$.

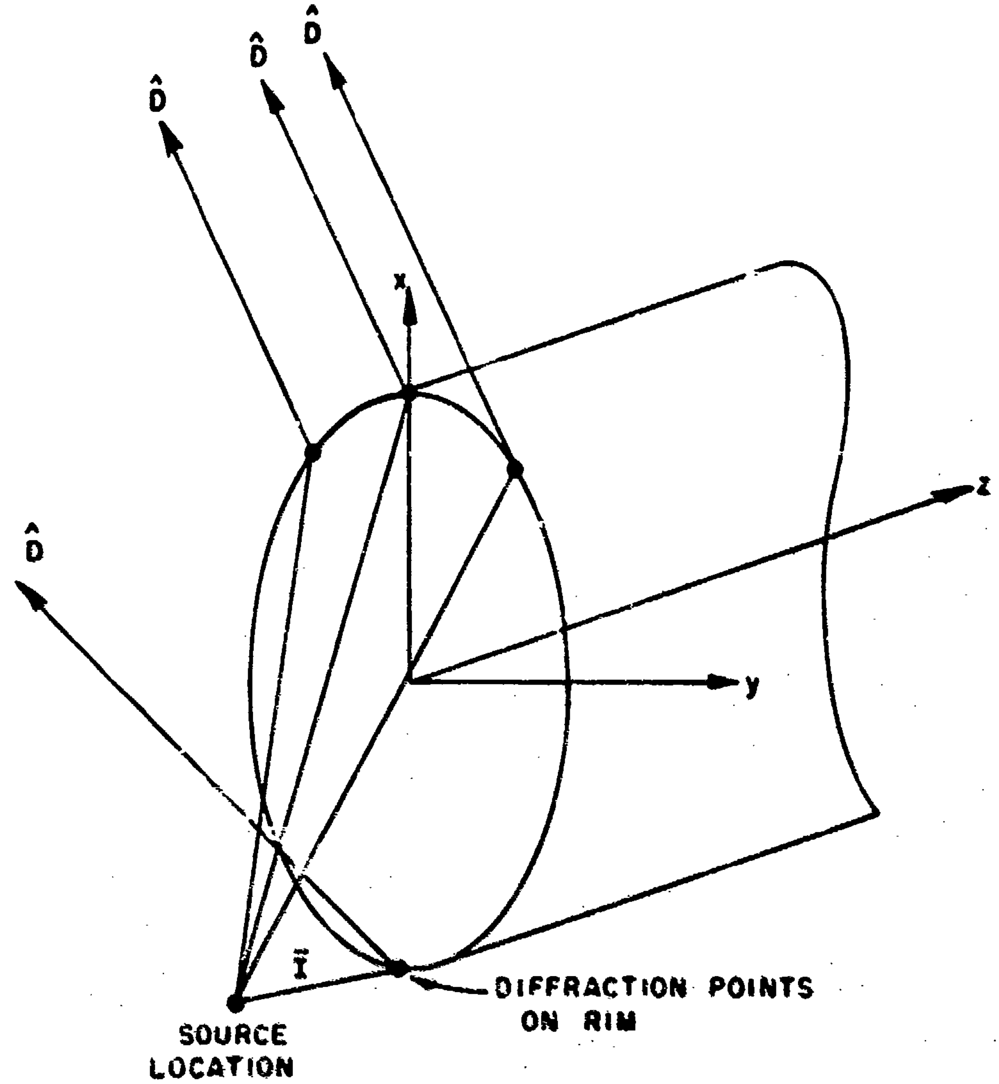
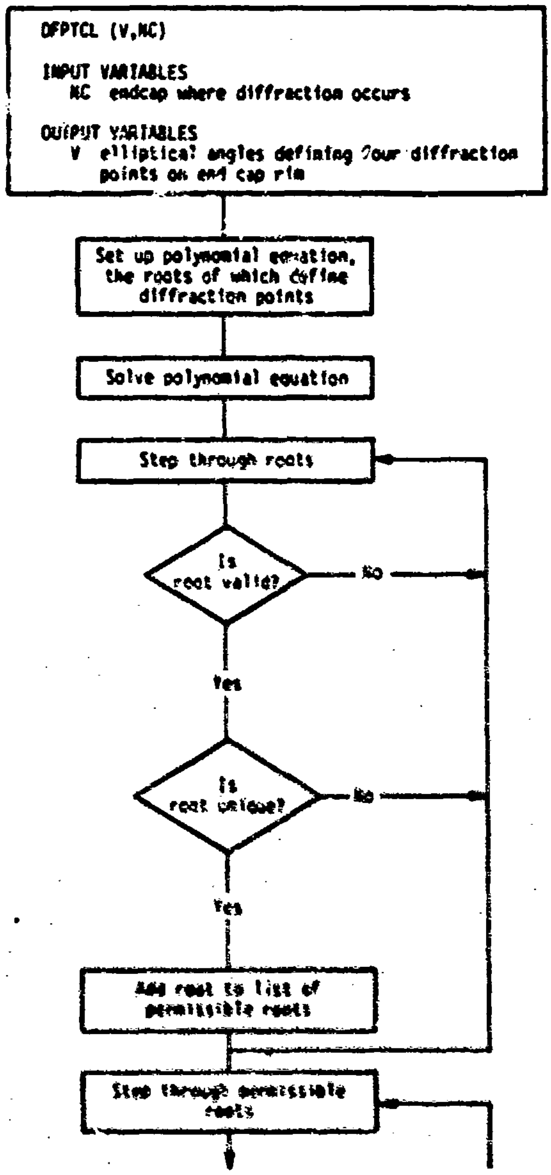


Figure 52-- Curved wedge diffraction points on rim of end cap of finite elliptic cylinder.

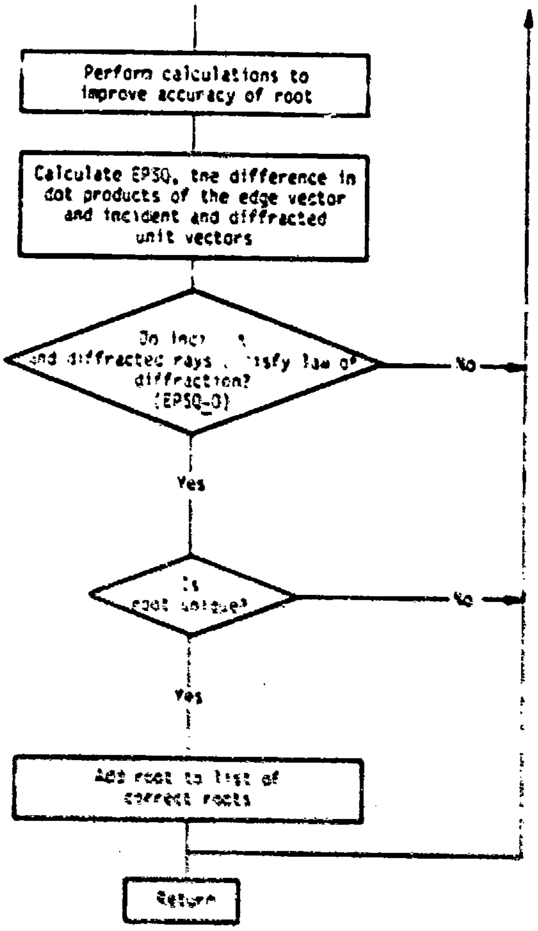
METHOD

An eighth order polynomial equation is used to solve for eight possible points on the end cap rim that can be diffraction points. These points are defined by elliptic angles in the local elliptic coordinate system for the end cap. The points are next integerized and sorted to remove duplicate points. The accuracy of the possible diffraction points are then improved by a first order Taylor series interpolation scheme. The details are given on pages 125-127 of Reference 1. The two to four correct diffraction points are verified by checking to see which of the remaining points satisfy the laws of diffraction.

FLOW DIAGRAM



DFPTCL (V,NC)
INPUT VARIABLES
NC endcap where diffraction occurs
OUTPUT VARIABLES
V elliptical angles defining four diffraction points on end cap rim
Set up polynomial equation, the roots of which define diffraction points
Solve polynomial equation
Step through roots
Is root valid?
No
Yes
Is root unique?
No
Yes
Add root to list of permissible roots
Step through permissible roots


Perform calculations to improve accuracy of root
Calculate EPSQ, the difference in dot products of the edge vector and incident and diffracted unit vectors
Do incident and diffracted rays satisfy law of diffraction? (EPSQ≅0)
No
Yes
Is root unique?
No
Yes
Add root to list of correct roots
Return

## SYMBOL DICTIONARY

| | |
|---|---|
| AE | HALF LENGTH OF END CAP (HALF LENGTH OF LINE CREATED BY INTERSECTION OF END CAP AND XZ PLANE) |
| C | COSINE OF VR |
| CC | POLYNOMIAL EQ. COEFFICIENTS |
| CV | COMPUTATIONAL VARIABLE |
| D4 | COMPUTATIONAL VARIABLE |
| DD | COMPUTATIONAL VARIABLE |
| DEEX<br>DEEY<br>DEEZ | X,Y,Z COMPONENTS OF VECTOR FROM DIFFRACTION POINT TO CENTER OF END CAP IN RCS |
| DEL | TEST VARIABLE |
| DEN1 | MAGNITUDE OF UNNORMALIZED EDGE UNIT VECTOR |
| DEN2 | DISTANCE FROM SOURCE TO IMPROVED DIFFRACTION POINT |
| DEN3 | LENGTH OF INCIDENT RAY VECTOR |
| DEN5 | COMPUTATIONAL VARIABLE |
| DM | X,Y,Z COMPONENTS OF UNIT VECTOR OF PROPAGATION DIRECTION IN END CAP COORDINATE SYSTEM |
| DOTQ1 | DOT PRODUCT OF EDGE VECTOR AND INCIDENT RAY |
| DOTQ2 | DOT PRODUCT OF EDGE VECTOR AND DIFFRACTED RAY |
| DSSX<br>DSSY<br>DSSZ | X,Y,Z COMPONENTS OF VECTOR TANGENT TO DIFFRACTION POINT IN END CAP PLANE IN RCS |
| DV | CHANGE IN ELL ANGLE V CALCULATED TO IMPROVE ACCURACY OF DIFFRACTION POINT |
| EEX<br>EEY<br>EEZ | X,Y,Z COMPONENTS OF RAY TANGENT TO DIFFRACTION POINT IN RCS |
| EPSQ | DIFFERENCE IN DOTQ1 AND DOTQ2 (ERROR TEST VARIABLE) |
| ERCS | (NOR A VARIABLE) ABBR. FOR ELLIPTICAL REFERENCE COORDINATE SYSTEM |
| EXQ<br>EYQ<br>EZQ | X,Y,Z COMPONENTS OF NORMALIZED EDGE UNIT VECTOR IN RCS |
| I | DO LOOP VARIABLE |
| IDEL | TEST VARIABLE |
| IV | ELL ANGLES DEFINING PERMISSABLE DIFFRACTION POINTS IN ERCS (IN DEGREES, ROUNDED OFF TO NEAREST INTEGER) |
| J | ELL ANGLE DEFINING DIFFRACTION POINT IN ERCS IN DEG. |
| K | DO LOOP VARIABLE |
| N | INDEX VARIABLE (ALSO NUMBER OF PERMISSABLE ROOTS) |
| NC | END CAP WHERE DIFFRACTION OCCURS |
| NCC | SIGN CHANGE VARIABLE |
| P | POLYNOMIAL EQ. VARIABLE |
| Q | POLYNOMIAL EQ. VARIABLE |
| QC | COMPLEX CONJ. OF Q |
| R | POLYNOMIAL EQ. VARIABLE |
| RC | COMPLEX CONJ. OF R |
| ROOT | ROOTS OF POLYNOMIAL EQ RETURNED FROM SUB. POLYRT |
| S | SINE OF ELL ANGLE V (ALSO POLY. EQ. VARIABLE) |
| SSX<br>SSY<br>SSZ | X,Y,Z COMPONENTS OF VECTOR INCIDENT ON EDGE IN RCS |
| SXQ<br>SYQ<br>SZQ | X,Y,Z COMPONENTS OF UNIT VECTOR OF PROPAGATION DIRECTION OF INCIDENT RAY IN RCS |
| V | ELL ANGLES DEFINING DIFFRACTION POINTS IN ERCS |
| VQ | ELL ANGLE DEFINING DIFFRACTION POINT (IMPROVED ACCURACY) |
| VR | ELL ANGLE DEFINING DIFFRACTION POINT |
| VT | ELL ANGLE DEFINING DIFFRACTION POINT (IMPROVED ACCURACY) IN DEGREES |
| XSM<br>YSM<br>ZSM | X,Y,Z COMPONENTS OF SOURCE LOCATION IN END CAP COORDINATE SYSTEM |

# CODE LISTING

```
C-----------------------------------------------------------------------
      SUBROUTINE DFPTCL(V,NC)
C!!!
C!!!  DETERMINES THE DIFFRACTION POINT ON THE CURVED
C!!!  EDGE OF THE ELLIPTIC CYLINDER END CAP
C!!!
      COMPLEX CC(9),ROOT(8),CV,Q,QC,R,RC
      DIMENSION IV(8),V(4),DM(3)
      COMMON/GEOMEL/A,B,ZC(2),SNC(2),CNC(2).CTC(2)
      COMMON/SORINF/XS(3),VXS(3,3)
      COMMON/PIS/PI,TPI,DPR,RPD
      COMMON/DIR/D(3),THSR,PHSR,SPS,CPS,STHS,CTHS
      NCC=NC
      IF(NC.GT.1) NCC=-1
      DO 10 I=1,8
      IV(I)=-1000
      IF(I.LE.4) V(I)=-1000.
   10 CONTINUE
      DM(1)=SNC(NC)*D(1)+CNC(NC)*D(3)
      DM(2)=D(2)
      DM(3)=-CNC(NC)*D(1)+SNC(NC)*D(3)
      XSM=SNC(NC)*XS(1)+CNC(NC)*(XS(3)-ZC(NC))
      YSM=XS(2)
      ZSM=-CNC(NC)*XS(1)+SNC(NC)*(XS(3)-ZC(NC))
      AE=A/SNC(NC)
C!!!  SET UP POLYNOMIAL EQUATION
      P=AE*AE-B*B
      IF(ABS(AE-B).LT.1.E-9) P=0.
      Q=CMPLX(AE*XSM,-B*YSM)
      QC=CONJG(Q)
      R=CMPLX(B*DM(2),AE*DM(1))
      RC=CONJG(R)
      S=AE*AE+B*B+2.*(XSM*XSM+YSM*YSM+ZSM*ZSM)
      CC(9)=P*(P+R*R)
      CC(8)=-4.*Q*(P+R*R)
      CC(7)=2.*(2.*Q*Q+S*R*R+P*R*RC)
      CC(6)=4.*(QC*(P-R*R)-2.*Q*R*RC)
      CC(5)=CMPLX(0.,0.)
      CC(5)=CC(5)+P*(R*R+RC*RC)-2.*(P*P+4.*Q*QC)+4.*S*R*RC
      CC(4)=CONJG(CC(6))
      CC(3)=CONJG(CC(7))
      CC(2)=CONJG(CC(8))
      CC(1)=CONJG(CC(9))
C!!!  SOLVE POLYNOMIAL EQUATION
      CALL POLYRT(8,CC,ROOT)
      N=0
C!!!  STEP THRU ROOTS
      DO 200 I=1,8
C!!!  CHECK TO SEE IF ROOT IS VALID
      RM=CABS(ROOT(I))
      IF(RM.LT.0.1) GO TO 200
      CV=DPR*CMPLX(0.,-1.)*CLOG(ROOT(I))
      VT=ABS(1.-RM)
      IF(VT.GT.0.1) GO TO 200
      IF(REAL(CV).GE.0.) J=REAL(CV)+.5
      IF(REAL(CV).LT.0.) J=REAL(CV)-.5
      IF(J.LT.0) J=J+360
      IF(J.GE.360) J=J-360
      IF(N.EQ.0) GO TO 151
      DO 150 K=1,N
      IDEL=IABS(J-IV(K))
C!!!  IS ROOT UNIQUE? IF SO ADD TO LIST OF PERMISSABLE ROOTS
      IF(IDEL.LE.1.OR.IDEL.GE.359) GO TO 200
  150 CONTINUE
  151 N=N+1
      IV(N)=J
```

```
 200  CONTINUE
      IF(N.EQ.Ø) GO TO 3Ø31
      J=Ø
C!!!  STEP THRU PERMISSABLE ROOTS
      DO 3ØØ I=1,N
C!!!  PERFORM CALCULATION TO IMPROVE ACCURACY OF ROOT
      VR=IV(I)*RPD
      S=SIN(VR)
      C=COS(VR)
      DSSX=-A*S
      DSSY=B*C
      DSSZ=-A*CTC(NC)*S
      DEEX=-A*C
      DEEY=-B*S
      DEEZ=-A*CTC(NC)*C
      SSX=A*C-XSM
      SSY=B*S-YSM
      SSZ=A*CTC(NC)*C-XS(3)+ZC(NC)
      DEN3=SQRT(SSX*SSX+SSY*SSY+SSZ*SSZ)
      EEX=DSSX
      EEY=DSSY
      EEZ=DSSZ
      DD=(EEX*SSX+EEY*SSY+EEZ*SSZ)/DEN3
      DEN5=DEN3*(EEX*DM(1)+EEY*DM(2)+EEZ*DM(3))-EEX*SSX-EEY*SSY-EEZ*SSZ
      D4=EEX*DSSX+EEY*DSSY+EEZ*DSSZ+DEEX*SSX+DEEY*SSY+DEEZ*SSZ
      D4=D4-DD*(EEX*DM(1)+EEY*DM(2)+EEZ*DM(3))
      D4=D4-DEN3*(DEEX*DM(1)+DEEY*DM(2)+DEEZ*DM(3))
      DV=DEN5*DPR/D4
      IF(ABS(DV).GT.2.) GO TO 3ØØ
      VT=IV(I)+DV
      VQ=VT*RPD
      S=SIN(VC)
      C=COS(VQ)
      DEN1=A*A*S*S+B*B*C*C+A*A*S*S*CTC(NC)*CTC(NC)
      DEN1=SQRT(DEN1)
      DEN2=(A*C-XSM)*(A*C-XSM)+(B*S-YSM)*(B*S-YSM)
      DEN2=SQRT(DEN2+(A*CTC(NC)*C-XS(3)+ZC(NC))
     2*(A*CTC(NC)*C-XS(3)+ZC(NC)))
      EXQ=-A*S/DEN1
      EYQ=B*C/DEN1
      EZQ=-A*CTC(NC)*S/DEN1
      SXQ=(A*C-XSM)/DEN2
      SYQ=(B*S-YSM)/DEN2
      SZQ=(A*CTC(NC)*C-XS(3)+ZC(NC))/DEN2
C!!!  CALCULATE EPSQ, THE DIFFERENCE IN DOT PRODUCTS OF THE EDGE
C!!!  VECTOR AND INC. AND DIF. PROPAGATION UNIT VECTORS
      DOTQ1=SXQ*EXQ+SYQ*EYQ+SZQ*EZQ
      DOTQ2=DM(1)*EXQ+DM(2)*EYQ+DM(3)*EZQ
      EPSQ=DOTQ1-DOTQ2
C!!!  DO INC. AND DIF. RAYS SATISFY LAW OF DIFFRACTION (EPSQ=Ø)
      IF(ABS(EPSQ).GT.1.E-3) GO TO 3ØØ
      IF(VT.GE.36Ø.) VT=VT-36Ø.
      IF(VT.LT.Ø.) VT=36Ø.+VT
      IF(J.EQ.0) GO TO 289
      DO 288 K=1,J
      DEL=ABS(VT-V(K))
C!!!  IS THE ROOT UNIQUE?  IF SO, ADD TO LIST OF CORRECT ROOTS
      IF(DEL.LT.Ø.5.OR.DEL.GT.359.5) GO TO 3ØØ
 288  CONTINUE
 289  J=J+1
      V(J)=VT
 3ØØ  CONTINUE
 3Ø31 RETURN
      END
```

## DFPTWD

### PURPOSE

To determine the diffraction point along the line tangent to edge ME of plate MP for given source location $\overline{XS}$ and diffracted ray direction $\hat{D}$.

### PERTINENT GEOMETRY

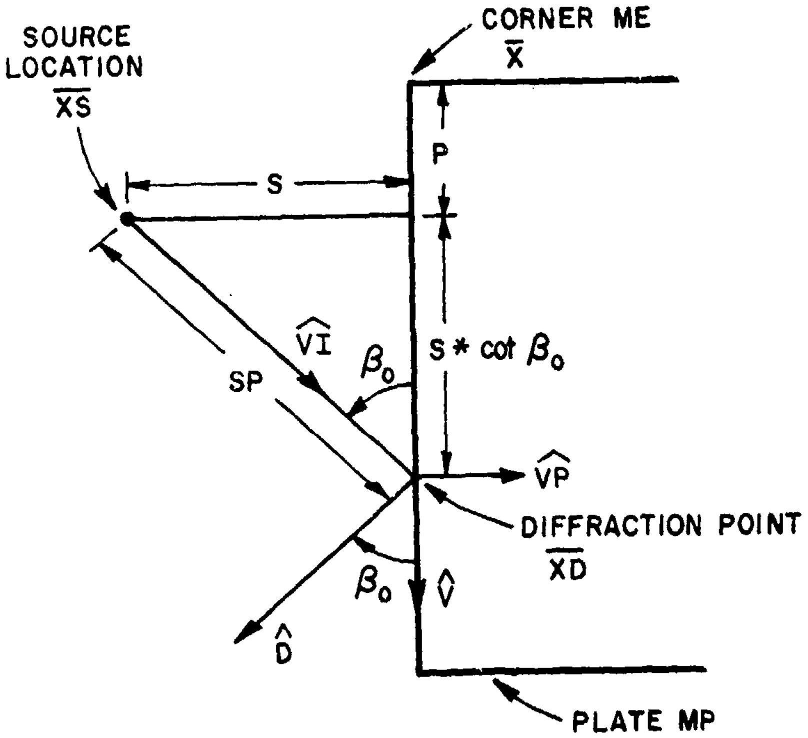


Figure 53--Geometry used in defining diffraction point on plate edge.

### METHOD

The diffraction point is found using similar triangles. Since $\cos\beta_0 = \hat{D}\cdot\hat{V}$ is known, then

$$\overline{XD} = \overline{X} + (S\cot\beta_0 + P)\hat{V}\ .$$

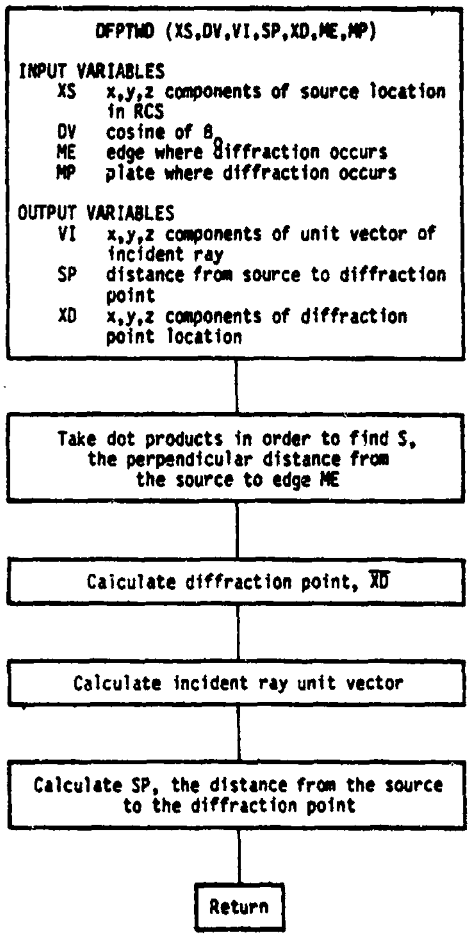
$$\overline{XS} = \hat{x}\,XS(1) + \hat{y}\,XS(2) + \hat{z}\,XS(3)$$

$$\overline{XD} = \hat{x}\,XD(1) + \hat{y}\,XD(2) + \hat{z}\,XD(3)$$

$$\overline{X} = \hat{x}\,X(MP,ME,1) + \hat{y}\,X(MP,ME,2) + \hat{z}\,X(MP,ME,3)$$

$$\hat{D} = \hat{x}\,D(1) + \hat{y}\,D(2) + \hat{z}\,D(3)$$

## FLOW DIAGRAM


DFPTWD (XS,DV,VI,SP,XD,ME,MP)
INPUT VARIABLES
XS x,y,z components of source location in RCS
DV cosine of β
ME edge where diffraction occurs
MP plate where diffraction occurs
OUTPUT VARIABLES
VI x,y,z components of unit vector of incident ray
SP distance from source to diffraction point
XD x,y,z components of diffraction point location
Take dot products in order to find S, the perpendicular distance from the source to edge ME
Calculate diffraction point, $\overline{XD}$
Calculate incident ray unit vector
Calculate SP, the distance from the source to the diffraction point
Return

## SYMBOL DICTIONARY

| | |
|---|---|
| CTB | COTANGENT OF BETA |
| DV | COSINE OF BETA |
| ME | EDGE WHERE DIFFRACTION OCCURS |
| MP | PLATE WHERE DIFFRACTION OCCURS |
| N | DO LOOP VARIABLE |
| P | DOT PRODUCT OF EDGE VECTOR AND VECTOR FROM CORNER ME TO SOURCE |
| S | PERPENDICULAR DISTANCE FROM SOURCE TO EDGE ME |
| SP | DISTANCE FROM SOURCE TO DIFFRACTION POINT |
| SX | VARIABLE USED TO CALCULATE S |
| VI | INCIDENT RAY UNIT VECTOR |
| XD | LOCATION OF DIFFRACTION POINT |
| XS | SOURCE LOCATION |

## CODE LISTING

```
C------------------------------------------------------------------
      SUBROUTINE DFPTWD(XS,DV,VI,SP,XD,ME,MP)
C!!!
C!!!  DETERMINATION OF THE DIFFRACTION POINT
C!!!
      DIMENSION XS(3),XD(3),VI(3)
      COMMON/GEOPLA/X(14,6,3),V(14,6,3),VP(14,6,3),VN(14,3)
     2,MEP(14),MPX
      CTB=DV/SQRT(1.-DV*DV)
      P=Ø.
      DO 1Ø N=1,3
   1Ø P=P+(XS(N)-X(MP,ME,N))*V(MP,ME,N)
      S=Ø.
      DO 2Ø N=1,3
      SX=XS(N)-X(MP,ME,N)-P*V(MP,ME,N)
   2Ø S=S+SX*SX
      S=SQRT(S)
      DO 3Ø N=1,3
   3Ø XD(N)=X(MP,ME,N)+(S*CTB+P)*V(MP,ME,N)
      SP=Ø.
      DO 4Ø N=1,3
      VI(N)=XD(N)-XS(N)
   4Ø SP=SP+VI(N)*VI(N)
      SP=SQRT(SP)
      DO 5Ø N=1,3
   5Ø VI(N)=VI(N)/SP
      RETURN
      END
```

DFRFPT

PURPOSE

To determine the ray path for a source ray which is diffracted off of a given edge on a given plate and then reflected in a given direction by the cylinder.

PERTINENT GEOMETRY

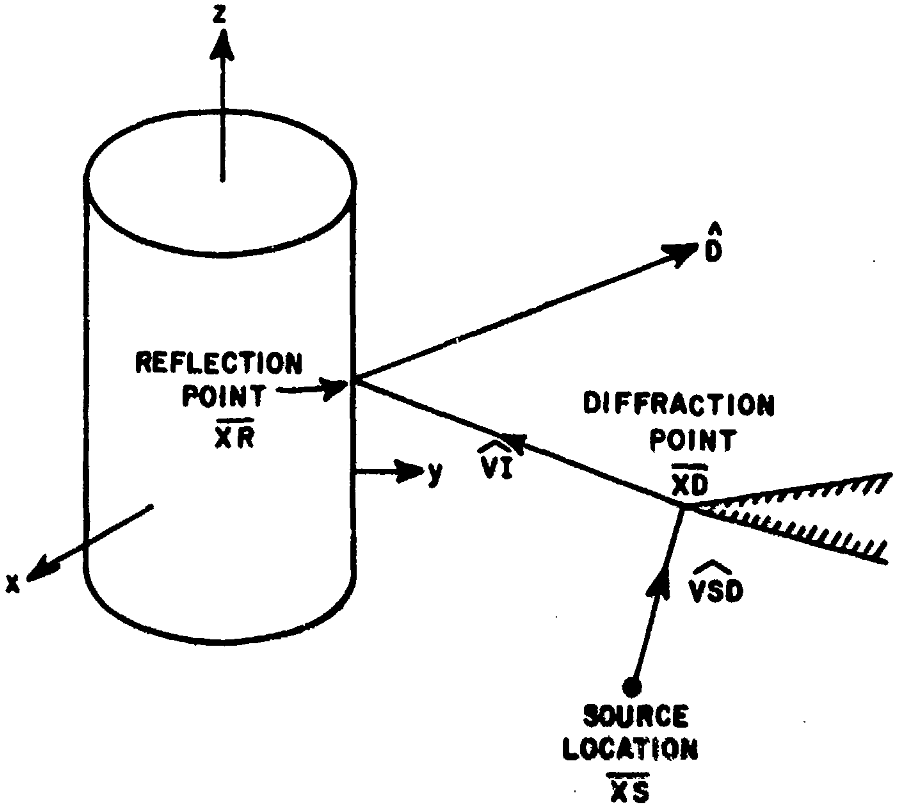
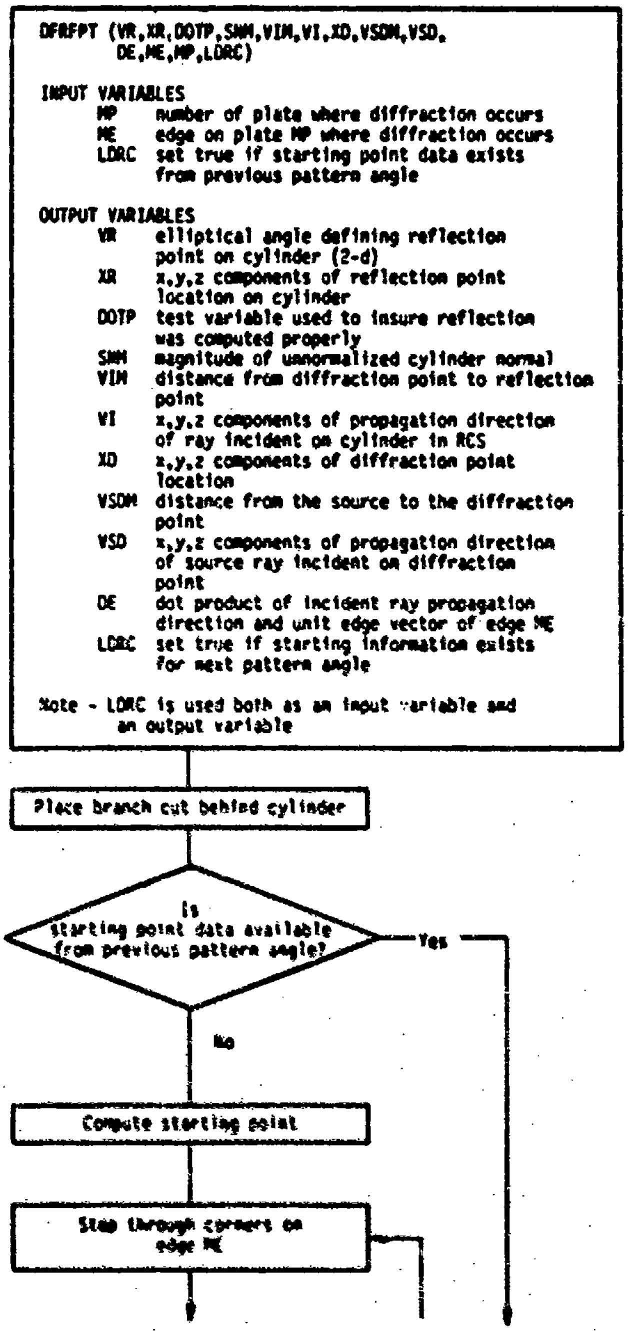

z
$\hat{D}$
REFLECTION POINT $\overline{XR}$
DIFFRACTION POINT $\overline{XD}$
y
$\widehat{VI}$
x
$\widehat{VSD}$
SOURCE LOCATION $\overline{XS}$

Figure 54--Ray diffracted by plate and then reflected by the cylinder.

METHOD

The diffraction point on a plate edge and the reflection point on an elliptic cylinder for a diffracted-reflected ray in a given observation direction are calculated via an iterative process. The equations are based on a first order Taylor series approximation to the equations governing the laws of reflection and diffraction. The details of the analysis are given on pages 155-161 of Reference 1. The iteration process follows the same basic scheme outlined in the write up for subroutine RFPTCL. The initial start up procedure for this subroutine is composed of defining a known reflection point which is taken to be on the rim of the finite cylinder closest to the plate edge under consideration and then determining the corresponding diffraction point on the plate edge. The details of this procedure are discussed on pages 161-163 of Reference 1.

## FLOW DIAGRAM



DFRFPT (VR,XR,DOTP,SNM,VIM,VI,XD,VSDM,VSD,
        DE,ME,MP,LDRC)

INPUT VARIABLES

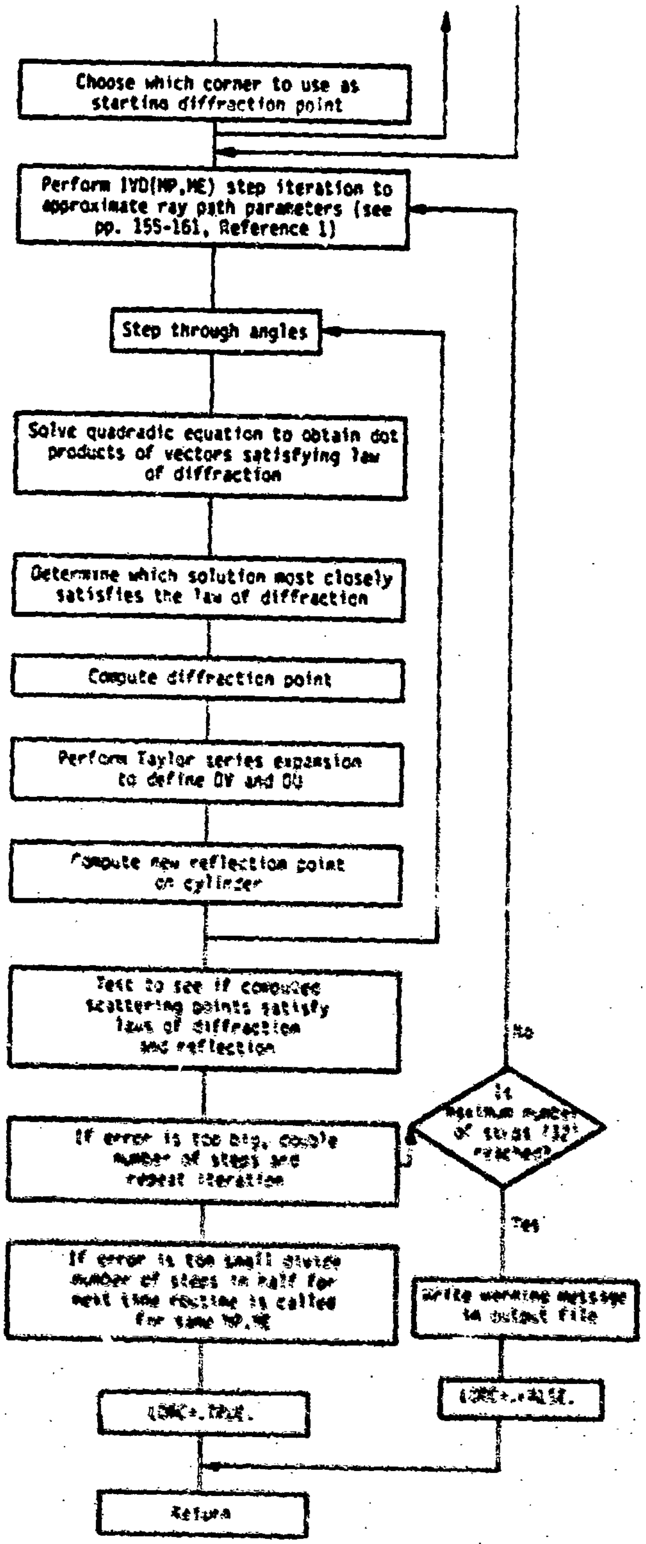
MP   number of plate where diffraction occurs
ME   edge on plate MP where diffraction occurs
LDRC set true if starting point data exists from previous pattern angle

OUTPUT VARIABLES

VR   elliptical angle defining reflection point on cylinder (2-d)
XR   x,y,z components of reflection point location on cylinder
DOTP test variable used to insure reflection was computed properly
SNM  magnitude of unnormalized cylinder normal
VIM  distance from diffraction point to reflection point
VI   x,y,z components of propagation direction of ray incident on cylinder in RCS
XD   x,y,z components of diffraction point location
VSDM distance from the source to the diffraction point
VSD  x,y,z components of propagation direction of source ray incident on diffraction point
DE   dot product of incident ray propagation direction and unit edge vector of edge ME
LDRC set true if starting information exists for next pattern angle

Note - LDRC is used both as an input variable and an output variable

Place branch cut behind cylinder
Is starting point data available from previous pattern angle?
Yes
No
Compute starting point
Step through corners on edge ME


Choose which corner to use as starting diffraction point
Perform IYD(NP,ME) step iteration to approximate ray path parameters (see pp. 155-161, Reference 1)
Step through angles
Solve quadradic equation to obtain dot products of vectors satisfying law of diffraction
Determine which solution most closely satisfies the law of diffraction
Compute diffraction point
Perform Taylor series expansion to define DV and DU
Compute new reflection point on cylinder
Test to see if computed scattering points satisfy laws of diffraction and reflection
If error is too big, double number of steps and repeat iteration
If error is too small divide number of steps in half for next time routine is called for same NP,ME
LOK=.TRUE.
Return
No
Is maximum number of steps (32) reached?
Yes
Write warning message in output file
LOK=.FALSE.

## SYMBOL DICTIONARY

| | |
|---|---|
| CSCE | DOT PRODUCT OF RAY FROM CORNER OF EDGE NE TO SOURCE AND EDGE UNIT VECTOR |
| DPSR | PHI ANGLE INCREMENT SIZE |
| DR | REFLECTED RAY PROPAGATION DIRECTION |
| DRP | X,Y COMPONENTS OF PHI POLARIZATION UNIT VECTOR FOR FIELD REFLECTED FROM CYLINDER IN RCS |
| DRT | X,Y,Z COMPONENTS OF THETA POLARIZATION UNIT VECTOR FOR FIELD REFLECTED FROM CYLINDER |
| DTSR | THETA ANGLE INCREMENT SIZE |
| DU | CHANGE IN UR FOR ONE ITERATION USING TAYLOR SERIES EXPANSION |
| DV | CHANGE IN VR FOR ONE ITERATION USING TAYLOR SERIES EXPANSION |
| ERC | ERROR DETECTION VARIABLE |
| FI | EQUATION GOVERNING THE LAW OF REFLECTION |
| FP | PARTIAL DERIVATIVE OF FI WITH RESPECT TO PHI |
| FT | PARTIAL DERIVATIVE OF FI WITH RESPECT TO THETA |
| FU | PARTIAL DERIVATIVE OF FI WITH RESPECT TO UR |
| FV | PARTIAL DERIVATIVE OF FI WITH RESPECT TO VR |
| GI | EQUATION GOVERNING THE LAW OF REFLECTION |
| GP | PARTIAL DERIVATIVE OF GI WITH RESPECT TO PHI |
| GT | PARTIAL DERIVATIVE OF GI WITH RESPECT TO THETA |
| GU | PARTIAL DERIVATIVE OF GI WITH RESPECT TO UR |
| GV | PARTIAL DERIVATIVE OF GI WITH RESPECT TO VR |
| IVD | NUMBER OF STEPS USED IN ITERATION |
| LDRC | SET TRUE IF STARTING POINT DATA IS AVAILABLE FROM PREVIOUS PATTERN ANGLE |
| PHCR | PHI COMPONENT OF REFLECTED RAY DIRECTION |
| PHLR | PHI COMPONENT OF REFLECTED RAY DIRECTION FROM PREVIOUS TIME DFRFPT WAS CALLED (OR PRESENT VALUE FOR NEXT TIME ROUTINE IS CALLED) |
| PHCRP | PHI ANGLE OF REFLECTED RAY DIRECTION IN ROTATED RCS SYSTEM (BRANCH CUT PLACED BEHIND CYL) |
| PHSPR | PHI ANGLE OF REFLECTED RAY DIRECTION IN ROTATED RCS SYSTEM (BRANCH CUT PLACED BEHIND CYLINDER) |
| SNPX | PARTIAL DERIVATIVE OF SNX WITH RESPECT TO VR |
| SNPY | PARTIAL DERIVATIVE OF SNY WITH RESPECT TO VR |
| SNX<br>SNY | X AND Y COMPONENTS OF NORMAL TO CYLINDER IN RCS COMPONENTS |
| STP | NUMBER OF STEPS USED IN ITERATION |
| THCR | THETA COMPONENT OF REFLECTED RAY DIRECTION |
| THLR | THETA COMPONENT OF REFLECTED RAY DIRECTION FROM PREVIOUS TIME RFDFPT WAS CALLED (OR FOR NEXT TIME ROUTINE IS CALLED) |
| URC | Z COMPONENT OF STARTING REFLECTION POINT LOCATION ON CYLINDER |
| VI | UNIT VECTOR OF INCIDENT RAY ON CYLINDER |
| VIU | PARTIAL DERIVATIVE OF VI WITH RESPECT TO UR |
| VIV | PARTIAL DERIVATIVE OF VI WITH RESPECT TO VR |
| VRC | ELL ANGLE DEFINING STARTING REFLECTION POINT ON CYLINDER |
| VSD | X,Y,Z COMPONENTS OF PROPAGATION VECTOR OF RAY FROM SOURCE TO DIFFRACTION POINT |
| XD | X,Y,Z COMPONENTS OF DIFFRACTION POINT LOCATION |
| XP | POINT ALONG LINE DRAWN THROUGH EDGE NE CLOSEST TO SOURCE |
| XR | X,Y,Z COMPONENTS OF REFLECTION POINT LOCATION ON CYLINDER |
| XRU | PARTIAL DERIVATIVE OF XR WITH RESPECT TO UR |
| XRV | PARTIAL DERIVATIVE OF XR WITH RESPECT TO VR |

## CODE LISTING

```
C-----------------------------------------------------------------------
      SUBROUTINE DFRFPT(VR,XR,DOTP,SNR,VIM,VI,XD,VSDM,VSD
     2,DE,ME,MP,LDRC)
C!!!
C!!!  DETERMINES THE RAY PATH FOR A DIFFRACTION FROM A PLATE THEN
C!!!  A REFLECTION FROM AN ELLIPTIC CYLINDER
C!!!
      DIMENSION DR(3),DMP(2),DRT(3),VI(3),VIM(3),VIU(3),VSD(3)
      DIMENSION XP(3),XR(3),XRP(3),XRV(3),XRU(3),XD(3)
      DIMENSION IVD(14,6),PHOR(14,6),THOR(14,6),VRO(14,6),URO(14,6)
      DIMENSION PHCRP(14,6)
      LOGICAL LDRC
      COMMON/GEOPL//X(14,6,3),V(14,6,3),VP(14,6,3),VN(14,3)
     2,MEP(14),MPX
      COMMON/SORINF/XS(3),VXS(3,3)
      COMMON/DIR/D(3),THSR,PHSR,SPHS,CPHS,STHS,CTHS
      COMMON/GEOMEL/A,B,ZC(2),SNC(2),CNC(2),CTC(2)
      COMMON/PDDCL/VDC(14,6),UDC(2),PDCR(14,6,2),TDCR(14,6,2)
     2,DTDC(14,6),PTDC(14,6,4),DDC(14,6,2)
      COMMON/BRNPH/PHBR(14,6)
      COMMON/PIS/PI,TPI,DPR,RPD
C!!!  PLACE BRANCH CUT BEHIND CYLINDER
      PHSPR=PHSR-PHBR(MP,ME)
      IF(PHSPR.GT.PI) PHSPR=PHSPR-TPI
      IF(PHSPR.LT.-PI) PHSPR=PHSPR+TPI
      CSCE=0.
      DO 2 M=1,3
    2 CSCE=CSCE+(XS(M)-X(MP,ME,M))*V(MP,ME,M)
      SSN=0.
      DO 3 M=1,3
      XP(M)=CSCE*V(MP,ME,M)+X(MP,ME,M)
    3 SSN=SSN+(XS(M)-XP(M))*(XS(M)-XP(M))
      SN=SQRT(SSN)
C!!!  IS STARTING POINT DATA AVAILABLE FROM PREVIOUS
C!!!  PATTERN ANGLE
      IF(LDRC) GO TO 40
C!!!  COMPUTE STARTING POINT
C!!!  STEP THRU CORNERS ON EDGE ME AND CHOOSE
C!!!  WHICH CORNER TO USE AS STARTING DIF. POINT
      CPDC=COS(PDCR(MP,ME,1))
      SPDC=SIN(PDCR(MP,ME,1))
      STDC=SIN(TDCR(MP,ME,1))
      DDC1=D(1)*CPDC*STDC+D(2)*SPDC*STDC+D(3)*DDC(MP,ME,1)
      CPDC=COS(PDCR(MP,ME,2))
      SPDC=SIN(PDCR(MP,ME,2))
      STDC=SIN(TDCR(MP,ME,2))
      DDC2=D(1)*CPDC*STDC+D(2)*SPDC*STDC+D(3)*DDC(MP,ME,2)
      MJ=1
      IF(DDC2.GT.DDC1) MJ=2
      PHOR(MP,ME)=PDCR(MP,ME,MJ)
      PHCRP(MP,ME)=PHOR(MP,ME)-PHBR(MP,ME)
      IF(PHCRP(MP,ME).GT.PI) PHCRP(MP,ME)=PHCRP(MP,ME)-TPI
      IF(PHCRP(MP,ME).LT.-PI) PHCRP(MP,ME)=PHCRP(MP,ME)+TPI
      THOR(MP,ME)=TDCR(MP,ME,MJ)
      VRO(MP,ME)=VDC(MP,ME)
      URO(MP,ME)=UDC(MJ)
      IVD(MP,ME)=1
C!!!  SPECIFY NUMBER OF STEPS IN ITERATION
   40 STP=IVD(MP,ME)
      IVDM=IVD(MP,ME)+1
      DPSP=(PHSPR-PHCRP(MP,ME))/STP
      DTS=(THSR-THOR(MP,ME))/STP
C!!!  SPECIFY STARTING POINT
      VR=VRO(MP,ME)
      UR=URO(MP,ME)
C!!!  THROUGH IVD STEP ITERATIONS TO NUMERICALLY
```

```
C!!!  COMPUTE THE DIFFRACTION AND REFLECTION POINTS.
C!!!  STEP THRU ANGLES
      DO 50 IV=1,IVDP
      PHCR=PHOR(MP,ME)+(IV-1)*DPSR
      THCR=THOR(MP,ME)+(IV-1)*DTSR
      CPCS=COS(PHCR)
      SPCS=SIN(PHCR)
      CTCS=COS(THCR)
      STCS=SIN(THCR)
      DR(1)=CPCS*STCS
      DR(2)=SPCS*STCS
      DR(3)=CTCS
      DRP(1)=-SPCS*STCS
      DRP(2)=CPCS*STCS
      DRT(1)=CPCS*CTCS
      DRT(2)=SPCS*CTCS
      DRT(3)=-STCS
      CSV=COS(VR)
      SNV=SIN(VR)
      SNX=B*CSV
      SNY=A*SNV
      SNPX=-B*SNV
      SNPY=A*CSV
      XR(1)=A*CSV
      XR(2)=B*SNV
      XR(3)=UH
      XRV(1)=-A*SNV
      XRV(2)=B*CSV
      XRV(3)=0.
      XRU(1)=0.
      XRU(2)=0.
      XRU(3)=1.
C!!!  SOLVE QUADRADIC EQUATION TO OBTAIN DOT PRODUCT
C!!!  OF VECTORS, SATISFYING LAW OF DIFFRACTION
      SSNP=0.
      DO 10 N=1,3
      XRP(N)=XR(N)-XP(N)
   10 SSRP=SSRP+XRP(N)*XRP(N)
      CRPV=XRP(1)*V(MP,ME,1)+XRP(2)*V(MP,ME,2)+XRP(3)*V(MP,ME,3)
      AA=(SSRP-SSH)*(SSRP-SSH)+4.*SSH*CRPV*CRPV
      BB=-2.*(SSH+SSRP)*CRPV*CRPV
      CC=CRPV*CRPV*CRPV*CRPV
      SQBAC=SQRT(BB*BB-4.*AA*CC)
C!!!  DETERMINE WHICH SOLUTION MOST CLOSELY SATISFIRS
C!!!  THE LAW OF DIFFRACTION
      ANSA=(-BB-SQBAC)/2./AA
      ANSB=(-BB+SQBAC)/2./AA
      CCIV=ANSA
      IF((ANSA.LT.0.).OR.(ANSA.GE.1.)) CCIV=ANSB
      CIV=-SQRT(CCIV)
      JCIV=0
   14 JCIV=JCIV+1
      VSDM=0.
      VIM=0.
      DO 11 N=1,3
C!!!  COMPUTE DIFFRACTION POINT
      XD(N)=XP(N)+SP*CIV*V(MP,ME,N)/SQRT(1.-CCIV)
      VSD(N)=XD(N)-XS(N)
      VSDM=VSDM+VSD(N)*VSD(N)
      VI(N)=XR(N)-XD(N)
   11 VIM=VIM+VI(N)*VI(N)
      VIM=SQRT(VIM)
      VSDM=SQRT(VSDM)
      CC=(VI(1)*V(MP,ME,1)+VI(2)*V(MP,ME,2)+VI(3)*V(MP,ME,3))/VIM
      FSD=(VSD(1)*V(MP,ME,1)+VSD(2)*V(MP,ME,2)+VSD(3)*V(MP,ME,3)
     2)/VSDM
```

```
      ERC=VE-ESD
      ERCB=ABS(ERC)
      IF(ERCB.LT.0.01) GO TO 15
      CIV=-CIV
      IF(JCIV.LT.2) GO TO 14
15    CONTINUE
      IF(IV.EQ.IVDP) GO TO 60
C!!!  PERFORM TAYLOR SERIES EXPANSION TO DEFINE DV AND DU
      CXRVE=XRV(1)*V(MP,ME,1)+XRV(2)*V(MP,ME,2)+XRV(3)*V(MP,ME,3)
      CXRVI=(XRV(1)*VI(1)+XRV(2)*VI(2)+XRV(3)*VI(3))/VIM
      CIVE=(CXRVE-CXRVI*CIV)/(VIM+SM/SQRT(1.-CCIV))
      CXRUE=XRU(1)*V(MP,ME,1)+XRU(2)*V(MP,ME,2)+XRU(3)*V(MP,ME,3)
      CXRUI=(XRU(1)*VI(1)+XRU(2)*VI(2)+XRU(3)*VI(3))/VIM
      CIUE=(CXRUE-CXRUI*CIV)/(VIM+SM/SQRT(1.-CCIV))
      DO 12 N=1,3
      VIV(N)=SM*CIVE*(1.+CCIV/(1.-CCIV))/SQRT(1.-CCIV)
      VIV(N)=XRV(N)-VIV(N)*V(MP,ME,N)
      VIU(N)=SM*CIUE*(1.+CCIV/(1.-CCIV))/SQRT(1.-CCIV)
12    VIU(N)=XRU(N)-VIU(N)*V(MP,ME,N)
      FV=(SNPX*VI(1)+SNX*VIV(1)+SNPY*VI(2)+SNY*VIV(2))*
     2(SNX*DR(2)-SNY*DR(1))
      FV=FV+(SNX*VI(1)+SNY*VI(2))*(SNPX*DR(2)-SNPY*DR(1))
      FV=FV+(SNPX*VI(2)+SNX*VIV(2)-SNPY*VI(1)-SNY*VIV(1))*
     2(SNX*DR(1)+SNY*DR(2))
      FV=FV+(SNX*VI(2)-SNY*VI(1))*(SNPX*DR(1)+SNPY*DR(2))
      FU=(SNX*DR(2)-SNY*DR(1))*(SNX*VIU(1)+SNY*VIU(2))+
     2(SNX*DR(1)+SNY*DR(2))*(SNX*VIU(2)-SNY*VIU(1))
      GV=DR(3)*(SNPX*VI(1)+SNX*VIV(1)+SNPY*VI(2)+SNY*VIV(2))
      GV=GV+VI(3)*(SNPX*DR(1)+SNPY*DR(2))
      GV=GV+VIV(3)*(SNX*DR(1)+SNY*DR(2))
      GU=DR(3)*(SNX*VIU(1)+SNY*VIU(2))+VIU(3)*(SNX*DR(1)+SNY*DR(2))
      FP=(SNX*VI(1)+SNY*VI(2))*(SNX*DRP(2)-SNY*DRP(1))+
     2(SNX*VI(2)-SNY*VI(1))*(SNX*DRP(1)+SNY*DRP(2))
      GP=VI(3)*(SNX*DRP(1)+SNY*DRP(2))
      GT=DRT(3)*(SNX*VI(1)+SNY*VI(2))+VI(3)*(SNX*DRT(1)+SNY*DRT(2))
      FI=(SNX*VI(1)+SNY*VI(2))*(SNX*DR(2)-SNY*DR(1))+
     2(SNX*DR(1)+SNY*DR(2))*(SNX*VI(2)-SNY*VI(1))
      GI=DR(3)*(SNX*VI(1)+SNY*VI(2))+VI(3)*(SNX*DR(1)+SNY*DR(2))
      DET=FU*GV-FV*GU
      DV=((FI*GU-GI*FU)+(GU*FP-FU*GP)*DPSR-FU*GT*DTSR)/DET
      DU=((GI*FV-FI*GV)+(FV*GP-GV*FP)*DPSR+FV*GT*DTSR)/DET
C!!!  COMPUTE NEW REFLECTION POINT ON CYLINDER
      UR=UR+DU
40    VR=VR+DV
50    CONTINUE
60    CONTINUE
C!!!  TEST TO SEE IF COMPUTED SCATTER POINTS SATISFY
C!!!  LAWS OF DIFFRACTION AND REFLECTION
      SNM=SQRT(SNX*SNX+SNY*SNY)
      SNX=SNX/SNM
      SNY=SNY/SNM
      DO 20 N=1,3
      VSD(N)=VSD(N)/VSDM
20    VI(N)=VI(N)/VIM
      SHAD=SNX*D(1)+SNY*D(2)
      SHADC=SNX*VI(1)+SNY*VI(2)
      ERC=SHAD+SHADC
      DCTP=.5*(SHAD-SHADC)
      ERCA=ABS(ERC)
      ERC=ERCA
      IF(ERCB.GT.ERC)ERC=ERCB
C!!!  IF ERROR IS VERY SMALL, CUT NUMBER OF ITERATIONS
C!!!  IN HALF FOR NEXT TIME ROUTINE IS CALLED
      IF(ERC.LT.0.01) GO TO 80
C!!!  IF ERROR IS TOO BIG, DOUPLE NUMBER OF INCREMENTS
C!!!  (UP TO 32) AND REPEAT ITERATION
```

```
      IF(IVD(MP,ME).GE.32) GO TO 7Ø
      IVD(MP,ME)=2*IVD(MP,ME)
      GO TO 4Ø
7Ø    CONTINUE
      WRITE(6,1) PHSR,THSR,MP,ME,VR,UR,ERCA,ERCB
1     FORMAT(' ERROR IN DFRFPT= ',2F12.6,2I5,4F12.6)
      LDRC=.FALSE.
      RETURN
8Ø    CONTINUE
      IF(ERC.GE.Ø.ØØ1) GO TO 9Ø
      IF(IVD(MP,ME).EQ.1) GO TO 9Ø
      IVD(MP,ME)=IVD(MP,ME)/2
9Ø    CONTINUE
C!!!  STORE PARAMETERS FOR NEXT TIME DFRFPT IS CALLED
      VRO(MP,ME)=VR
      URO(MP,ME)=UR
      PHOR(MP,ME)=PHSR
      PHORP(MP,ME)=PHSPR
      THOR(MP,ME)=THSR
      IF(.NOT.LDRC) IVD(MP,ME)=1
      LDRC=.TRUE.
      RETURN
      END
```

## DI

### PURPOSE

To calculate the incident part or the reflection part of the wedge diffraction coefficient or the corner diffraction coefficient.

### METHOD

This subroutine computes either the incident part or the reflection part of the wedge or corner diffraction coefficient. The uniform Geometrical Theory of Diffraction [4] has been used to derive these terms. For wedge diffraction the coefficient is given as

$$DI(R,\beta,\sin\beta_o,n) = \frac{-e^{-j\pi/4}}{2n\sqrt{2\pi k}\,\sin\beta_o}\left\{\cot\left(\frac{\pi+\beta}{2n}\right) F\,[kRa^{+}(\beta)] + \cot\left(\frac{\pi-\beta}{2n}\right) F\,[kRa^{-}(\beta)]\right\},$$

where

$$\beta = \begin{cases} \phi-\phi', & \text{for the incident case} \\ \phi+\phi', & \text{for the reflection case,} \end{cases}$$

$$a^{\pm}(\beta) = 2\cos^2\left(\frac{2n\pi N^{\pm}-\beta}{2}\right),$$

in which $N^{\pm}$ are the integers which most nearly satisfy the equations

$$2\pi nN^{+}-(\beta) = \pi$$

$$2\pi nN^{-}-(\beta) = -\pi,$$

$F(x)$ is the transition function,

and

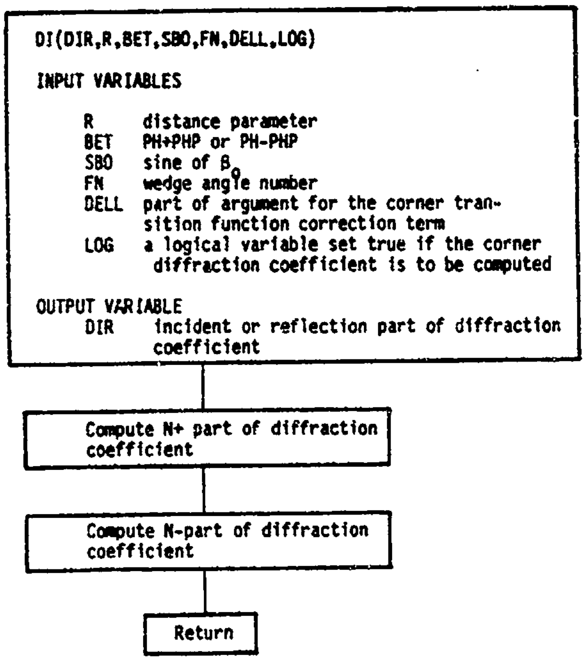
$n$ is the wedge number (FN).

For the corner diffracted term (LOG=.TRUE.), the coefficient is given as [9]:

$$DI(R,\beta,\sin\beta_o,n,R_c) = \frac{-e^{-j\pi/4}}{2n\sqrt{2\pi k}\,\sin\beta_o}\left\{\cot\left(\frac{\pi+\beta}{2n}\right) F\,[kRa^{+}(\beta)] \times \left| F\left[\frac{Ra^{+}(\beta)/\lambda}{kR_c a(\pi+\beta_o-\beta_c)}\right]\right| + \cot\left(\frac{\pi-\beta}{2n}\right) F\,[kRa^{-}(\beta)]\left| F\left[\frac{Ra^{-}(\beta)/\lambda}{kR_c a(\pi+\beta_o-\beta_c)}\right]\right|\right\},$$

where $R_c$ is the corner distance parameter and $\beta_c$ is the theta type angle measured from the corner. An illustration of the geometry is given in Figure 55.

## FLOW DIAGRAM



DI(DIR,R,BET,SBO,FN,DELL,LOG)

INPUT VARIABLES

| | |
|---|---|
| R | distance parameter |
| BET | PH+PHP or PH-PHP |
| SBO | sine of $\beta_o$ |
| FN | wedge angle number |
| DELL | part of argument for the corner transition function correction term |
| LOG | a logical variable set true if the corner diffraction coefficient is to be computed |

OUTPUT VARIABLE

| | |
|---|---|
| DIR | incident or reflection part of diffraction coefficient |

Compute N+ part of diffraction coefficient

Compute N-part of diffraction coefficient

Return

## SYMBOL DICTIONARY

| | |
|---|---|
| A | ANGULAR FUNCTION FOR TRANSITION FUNCTION |
| ANG | BET IN RADIANS |
| BOTL | ARGUMENT OF TRANSITION FUNCTION |
| C | REAL PART OF FRESNEL INTEGRAL |
| COM | CONSTANT FOR DIFFRACTION COEFFICIENT |
| COTA | COTANGENT TIMES THE SQUARE ROOT OF THE A FUNCTION |
| DEL | CORNER PART OF ARGUMENT FOR THE CORNER TRANSITION FUNCTION CORRECTION TERM |
| DELU | INVERSE OF DEL |
| DEM | 4*PI*FN*SIN(BO) |
| DN | INTEGER WHICH MOST NEARLY SATISFIES THE EQUATION, 2*PI*FN*DN-BET=PI OR -PI |
| DNS | COMPUTATIONAL VARIABLE |
| EX | CEXP(J*K*R*A) |
| FA | TRANSITION FUNCTION WITHOUT SQRT(A) |
| N | COMPUTATIONAL VARIABLE |
| RAG | ARGUMENT OF COTANGENT TERM |
| S | IMAGINARY PART OF FRESNEL INTEGRAL |
| SGN | SIGN OF DNS |
| SQR | SQRT(2*PI*R) |
| TS | ABSOLUTE VALUE OF TSIN |
| TSIN | SINE OF ARGUMENT OF COTANGENT TERM |
| UNPI | N- COMPONENT OF DI |
| UPPI | N+ COMPONENT OF DI |

CODE LISTING

```
C--------------------------------------------------------------------
      SUBROUTINE DI(DIR,R,BET,SBO,FN,DELL,LOG)
C!!!
C!!!  INCIDENT (BET=PH-PHP) OR REFLECTED (BET=PH+PHP)
C!!!  PART OF WEDGE DIFFRACTION COEFFICIENT
C!!!
      LOGICAL LOG,LDEBUG,LTEST
      COMMON/TEST/LDEBUG,LTEST
      COMPLEX FFCT,TOP,COM,EX,UPPI,UNPI,FA,DIR
      COMMON/TOPD/TOP
      COMMON/PIS/PI,TPI,DPR,RPD
      IF (LDEBUG) WRITE (6,11)
   11 FORMAT (/,' DEBUGGING DI SUBROUTINE')
      DEL=DELL
      IF(ABS(DEL).LT.1.E-10) DEL=SIGN(1.E-10,DEL)
      IF(LOG)DELU=1./DEL
      ANG=BET*RPD
      DEM=2.*TPI*FN*SBO
      COM=TOP/DEM
      SQR=SQRT(TPI*R)
C!!!  N+ PART OF DIFFRACTION COEFFICIENT
      DNS=(PI+ANG)/(2.0*FN*PI)
      SGN=SIGN(1.,DNS)
      N=IFIX(ABS(DNS)+0.5)
      DN=SGN*FLOAT(N)
      A=ABS(1.0+COS(ANG-2.0*FN*PI*DN))
      BOTL = 2.0*SQRT(ABS(R*A))
      EX=CEXP(CMPLX(0.0,TPI*R*A))
      CALL FRNELS (C,S,BOTL)
      C=SQRT(PI/2.0)*(0.5-C)
      S= SQRT(PI/2.0)*(S-0.5)
      FA=CMPLX(0.,2.)*SQR*EX*CMPLX(C,S)
      RAG=(PI+ANG)/(2.0*FN)
      TSIN=SIN(RAG)
      TS=ABS(TSIN)
      IF(TS.GT.1.E-5) GO TO 442
      COTA=-SQRT(2.0)*FN*SIN(ANG/2.0-FN*PI*DN)
      IF(COS(ANG/2.0-FN*PI*DN).LT.0.0) COTA=-COTA
      GO TO 443
  442 COTA=SQRT(A)*COS(RAG)/TSIN
  443 UPPI=COM*COTA*FA
      IF(LOG)UPPI=UPPI*BABS(FFCT(R*A*DELU))
      IF (LDEBUG) WRITE (6,*) DN,A,FA,UPPI
C!!!  N- PART OF DIFFRACTION COEFFICIENT
      DNS=(-PI+ANG)/(2.0*FN*PI)
      SGN=SIGN(1.,DNS)
      N=IFIX(ABS(DNS)+0.5)
      DN=SGN*FLOAT(N)
      A=ABS(1.0+COS(ANG-2.0*FN*PI*DN))
      BOTL = 2.0*SQRT(ABS(R*A))
      EX=CEXP(CMPLX(0.0,TPI*R*A))
      CALL FRNELS (C,S,BOTL)
      C=SQRT(PI/2.0)*(0.5-C)
      S= SQRT(PI/2.0)*(S-0.5)
      FA=CMPLX(0.,2.)*SQR*EX*CMPLX(C,S)
      RAG=(PI-ANG)/(2.0*FN)
      TSIN=SIN(RAG)
      TS=ABS(TSIN)
      IF(TS.GT.1.E-5) GO TO 542
      COTA= SQRT(2.0)*FN*SIN(ANG/2.0-FN*PI*DN)
      IF(COS(ANG/2.0-FN*PI*DN).LT.0.0) COTA=-COTA
      GO TO 123
  542 COTA=SQRT(A)*COS(RAG)/TSIN
  123 UNPI=COM*COTA*FA
      IF(LOG)UNPI=UNPI*BABS(FFCT(R*A*DELU))
      IF (LDEBUG) WRITE (6,*) DN,A,FA,UNPI
      DIR=UPPI+UNPI
      IF (.NOT.LTEST) GO TO 2
      WRITE (6,1)
 1    FORMAT (/,' TESTING DI SUBROUTINE')
      WRITE (6,*) DIR,R,BET
      WRITE (6,*) SBO,FN
 2    RETURN
      END
```

## DIFPLT

### PURPOSE

To calculate the far zone electric field for a source ray which is diffracted off of a given edge on a given plate.

### PERTINENT GEOMETRY

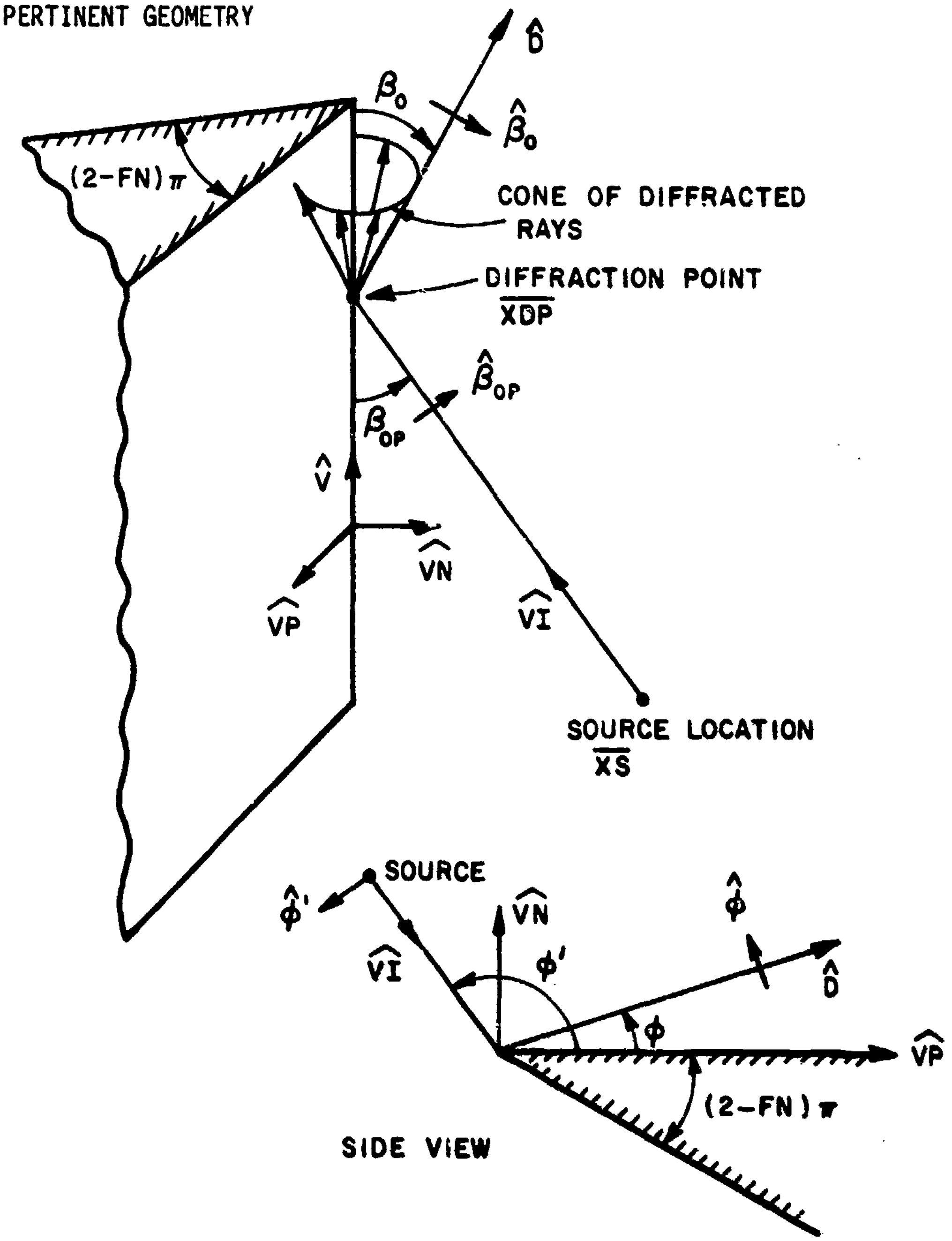


Figure 55--Edge diffraction geometry.

$$\hat{\beta}_o = \hat{x}\ BO(1) + \hat{y}\ BO(2) + \hat{z}\ BO(3)$$

$$\hat{\beta}_{op} = \hat{x}\ BOP(1) + \hat{y}\ BOP(2) + \hat{z}\ BOP(3)$$

$$\hat{\phi} = \hat{x}\ PH(1) + \hat{y}\ PH(2) + \hat{z}\ PH(3)$$

$$\hat{\phi}' = \hat{x}\ PHO(1) + \hat{y}\ PHO(2) + \hat{z}\ PHO(3)$$

$$\phi' = PSOR$$

$$\phi = PSR$$

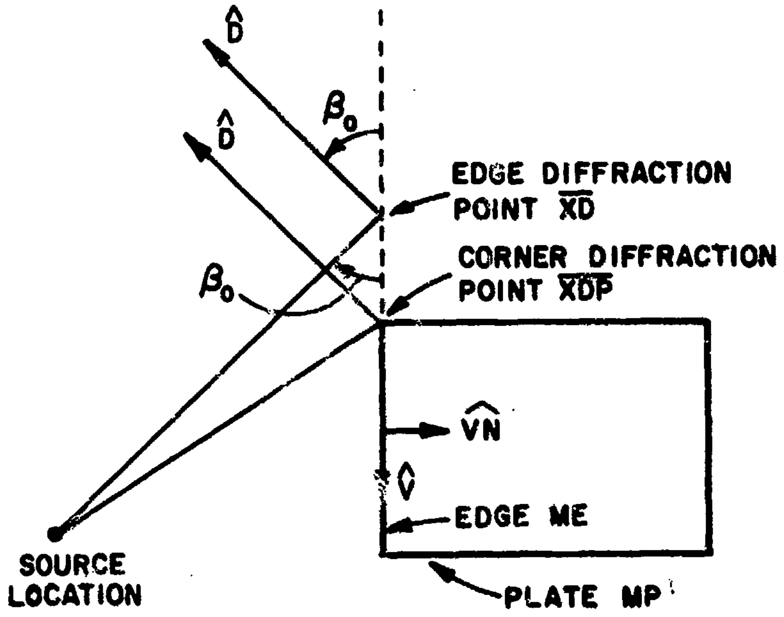


Figure 56--Corner diffraction geometry.

METHOD

The diffracted fields from the edges of the plates are calculated by using the Geometrical Theory of Diffraction [4]. The diffracted field in the far zone has the form [4]

$$\overline{E}^d = \overline{E}^i(Q_E) \cdot \overline{\overline{D}}_E(s',\phi,\phi',\beta_o,FN)\sqrt{s'}\ \frac{e^{-jks}}{s} ,$$

where $Q_E$ is the diffraction point. The incident field can be written in the form

$$\overline{E}^i(Q_E) = \left[EIPR\ \hat{\phi}' + EIPL\ \hat{\beta}_{opJ}\right]\frac{e^{-jks'}}{s'} .$$

The diffraction coefficient can be written as:

$$\overline{\overline{D}}_E(s',\phi,\phi',\beta_o,FN) = -DS\ \hat{\beta}_{op}\hat{\beta}_o - DH\ \hat{\phi}'\hat{\phi}\ .$$

The slope diffracted field in the far zone has the form[10]

$$E^{s.d.} = \frac{1}{jk\ \sin\beta_o} \frac{\partial \overline{E}^i(Q_E)}{\partial n'} \cdot \frac{\partial \overline{\overline{D}}_E}{\partial \phi'} \sqrt{s'}\ \frac{e^{-jks}}{s}$$

where $\frac{\partial \overline{E}^i}{\partial n} = \frac{1}{s'\ \sin\beta_o} \frac{\partial \overline{E}^i}{\partial \phi'}$. The incident slope field can be written in the form $\frac{\partial \overline{E}^i}{\partial n} = [EIPRP\hat{\phi}' + EIPLP\hat{\beta}_{op}]\frac{e^{-jks'}}{s'^2}$ where EIPRP and EIPLP are computed in subroutine SOURCP. The corner and slope corner diffracted fields have similar form[9] and are included if the logical variables LSLOPE and LCORNR are set true. The edge and slope fields are combined and the phase is referred to the reference coordinate system origin by the factor $e^{jkD\cdot\overline{XDP}}$. The form of the field is therefore given by

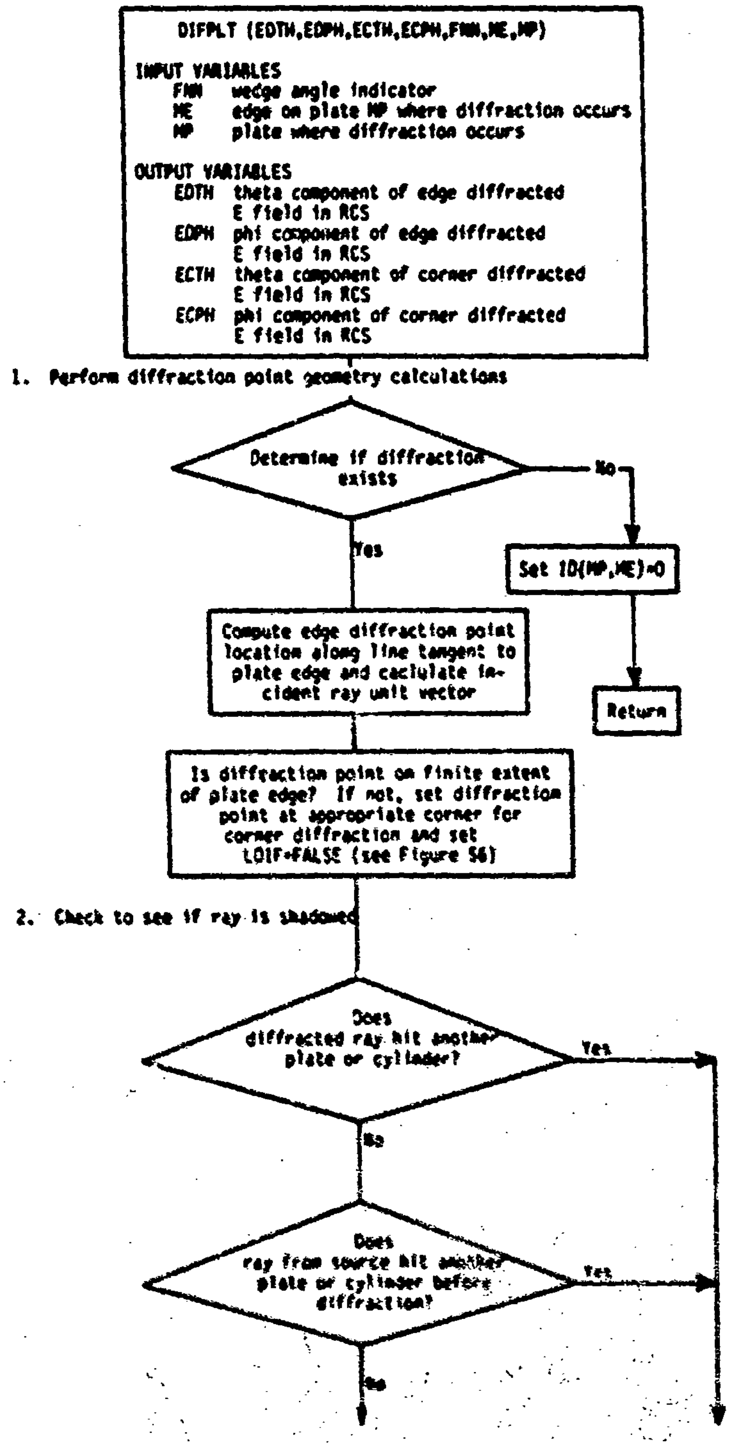
$$\overline{E}^d = W_m(EDTH\hat{\theta} + EDPH\hat{\phi})\ \frac{e^{-jkR}}{R}\ .$$

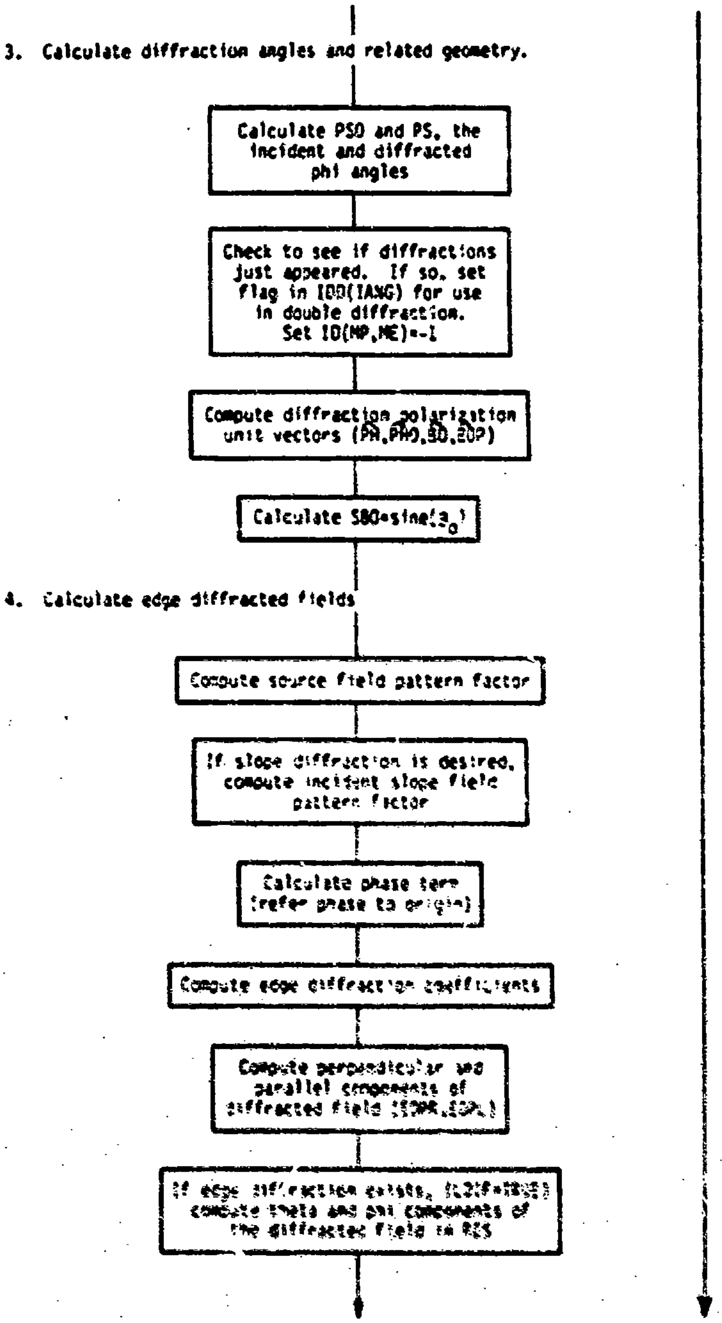
Similarly the corner and slope corner diffracted field is given by

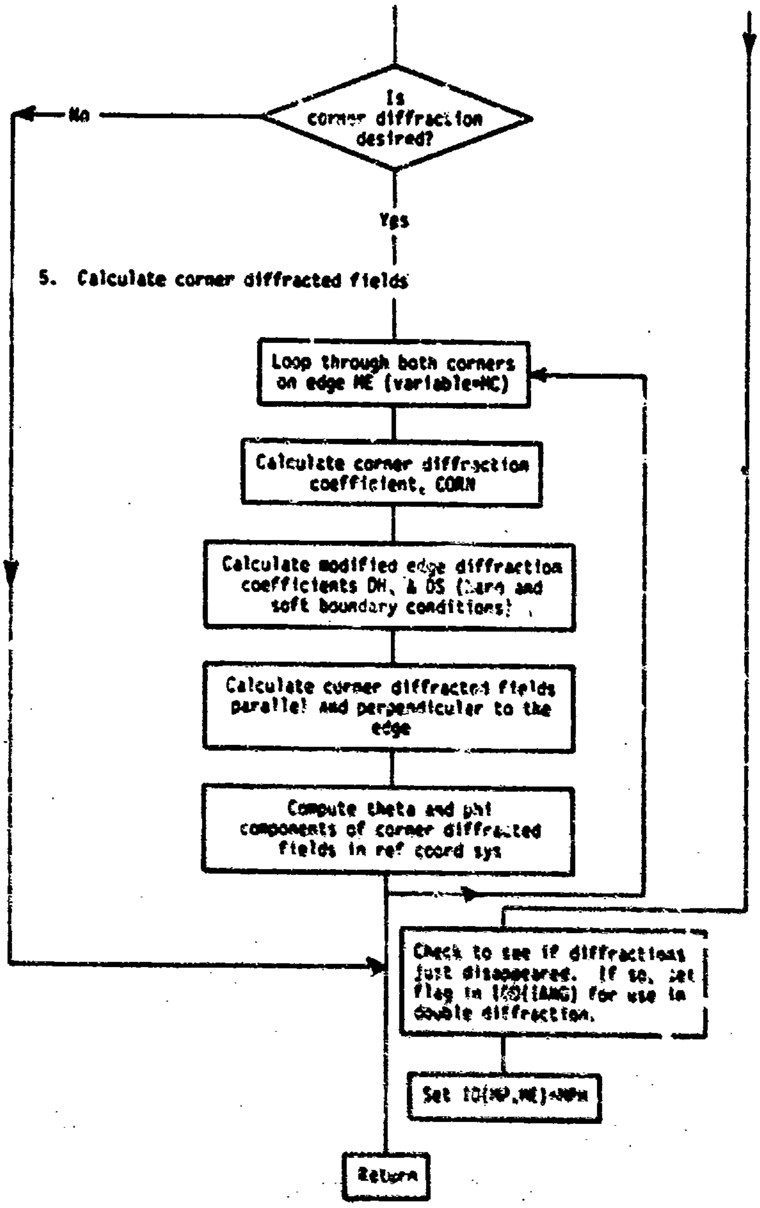
$$\overline{E}^c = W_m(ECTH\hat{\theta} + ECPH\hat{\phi})\ \frac{e^{-jkR}}{R}\ ,$$

where the factor $\frac{e^{-jkR}}{R}$ and the source weight ($W_m$) are added elsewhere in the code.

# FLOW DIAGRAM


DIFPLT (EDTH,EDPH,ECTH,ECPH,FNN,ME,MP)
INPUT VARIABLES
FNN wedge angle indicator
ME edge on plate MP where diffraction occurs
MP plate where diffraction occurs
OUTPUT VARIABLES
EDTH theta component of edge diffracted E field in RCS
EDPH phi component of edge diffracted E field in RCS
ECTH theta component of corner diffracted E field in RCS
ECPH phi component of corner diffracted E field in RCS
1. Perform diffraction point geometry calculations
Determine if diffraction exists
No
Set ID(MP,ME)=0
Return
Yes
Compute edge diffraction point location along line tangent to plate edge and caclulate incident ray unit vector
Is diffraction point on finite extent of plate edge? If not, set diffraction point at appropriate corner for corner diffraction and set LDIF=FALSE (see Figure 56)
2. Check to see if ray is shadowed
Does diffracted ray hit another plate or cylinder?
Yes
No
Does ray from source hit another plate or cylinder before diffraction?
Yes
No


3. Calculate diffraction angles and related geometry.
Calculate PSO and PS, the incident and diffracted phi angles
Check to see if diffractions just appeared. If so, set flag in IDD(IANG) for use in double diffraction. Set ID(MP,ME)=-1
Compute diffraction polarization unit vectors (PH,PHO,SD,SDP)
Calculate SBO=sine($\beta_o$)
4. Calculate edge diffracted fields
Compute source field pattern factor
If slope diffraction is desired, compute incident slope field pattern factor
Calculate phase term (refer phase to origin)
Compute edge diffraction coefficients
Compute perpendicular and parallel components of diffracted field (EDPR,EDPL)
If edge diffraction exists, (LDIF=TRUE) compute theta and phi components of the diffracted field in RCS


Is corner diffraction desired?
No
Yes
5. Calculate corner diffracted fields
Loop through both corners on edge NE (variable=NC)
Calculate corner diffraction coefficient, CORN
Calculate modified edge diffraction coefficients DH, & DS (hard and soft boundary conditions)
Calculate corner diffracted fields parallel and perpendicular to the edge
Compute theta and phi components of corner diffracted fields in ref coord sys
Check to see if diffractions just disappeared. If so, set flag in ICD(IANG) for use in double diffraction.
Set ID(NP,NE)=NPH
Return

## SYMBOL DICTIONARY

| | |
|---|---|
| ADN | DOT PRODUCT OF VECTOR FROM PLATE MP TO THE SOURCE AND THE PLATE UNIT NORMAL |
| AFN | WEDGE ANGLE NUMBER |
| BDEL | VARIABLE USED TO EXPAND DIFFRACTION ANGLE RANGE IF CORNER DIFFRACTION IS USED |
| BDHI | UPPER LIMIT FOR BD, THE COSINE OF THE DIFFRACTION ANGLE BETA |
| BDLOW | LOWER LIMIT FOR BD, THE COSINE OF THE DIFFRACTION ANGLE BETA |
| BETN | DIFFERENCE IN DIFFRACTED AND INCIDENT PHI ANGLES |
| BETP | SUM OF DIFFRACTED AND INCIDENT PHI ANGLES |
| BO | DIFFRACTED FIELD BETA POLARIZATION UNIT VECTOR (IN EDGE FIXED COORDIATE SYSTEM) IN RCS COMPONENTS |
| BOP | INCIDENT FIELD BETA POLARIZATION UNIT VECTOR (IN EDGE FIXED COORD SYS) IN RCS COMPONENTS |
| CNP | COSINE OF HALF WEDGE ANGLE |
| CORN | CORNER DIFFRACTION COEFFICIENT |
| CPH | COSINE OF PSR |
| CPHC | COSINE OF PSCR |
| CTH | COSINE OF THR |
| CTHP | COSINE OF THPR |
| DEL | PARAMETER USED IN TRANSITION FUNCTION |
| DH | DIFFRACTION COEF. FOR HARD BOUNDARY CONDITION |
| DHIT | DISTANCE FROM SOURCE TO NEAREST HIT (FROM SUBS. PLAINT OR CYLINT) |
| DPH | SLOPE DIFFRACTION COEFFICIENT FOR HARD BOUNDARY CONDITION |
| DPS | SLOPE DIFFRACTION COEFFICIENT FOR SOFT BOUNDARY CONDITION |
| DS | DIFFRACTION COEF. FOR SOFT BOUNDARY CONDITION |
| DV | DOT PRODUCT OF EDGE VECTOR AND PROPAGATION DIRECTION UNIT VECTOR,D WHICH IS THE COSINE OF BETA |
| ECBI | EDGE DIFFRACTION COEFFICIENT (FROM SUB. DI) FOR INCIDENT DIFFRACTED FIELD MODIFIED FOR CORNER DIFFRACTION |
| ECBR | EDGE DIFFRACTION COEFFICIENT (FROM SUB. DI) FOR REFLECTED DIFFRACTED FIELD MODIFIED FOR CORNER DIFFRACTION |
| ECPH | PHI COMPONENT OF CORNER DIFFRACTED E-FIELD |
| ECTH | THETA COMPONENT OF CORNER DIFFRACTED E-FIELD |
| EDPH | PHI COMPONENT OF EDGE DIFFRECTED E-FIELD |
| EDPL | COMPONENT OF DIFFRACTED FIELD PARALLEL TO THE EDGE |
| EDPR | COMPONENT OF DIFFRACTED FIELD PERPENDICULAR TO THE EDGE |
| EDTH | THETA COMPONENT OF EDGE DIFFRACTED E-FIELD |
| EF | THETA COMPONENT OF CORNER DIFFRACTED FIELD IN RCS |
| EG | PHI COMPONENT OF CORNER DIFFRACTED FIELD IN RCS |
| EIPL | COMPONENT OF INCIDENT FIELD PARALLEL TO THE EDGE |
| EIPLP | PATTERN FACTOR FOR COMPONENT OF SOURCE (INCIDENT) SLOPE FIELD PARALLEL TO THE EDGE |
| EIPR | COMPONENT OF INCIDENT FIELD PERPENDICULAR TO THE EDGE |
| EIPRP | PATTERN FACTOR FOR COMPONENT OF SOURCE (INCIDENT) SLOPE FIELD PERPENDICULAR TO THE EDGE |
| EIX<br>EIY<br>EIZ | SOURCE PATTERN FACTORS FOR X,Y, AND Z COMPONENTS OF INCIDENT E FIELD |
| EXPH | COMPLEX PHASE TERM (REFER PHASE TO RCS. ORIGIN) |
| FN | WEDGE ANGLE NUMBER |
| FNN | WEDGE ANGLE INDICATOR |
| FNP | ANGLE EXTERIOR TO WEDGE IN DEGREES |
| GAM | DOT PRODUCT OF THE PROPAGATION DIRECTION AND THE VECTOR FROM THE REF COORD SYS ORIGIN TO THE DIFFRACTION POINT |
| ID | ARRAY OF FLAGS INDICATING WHETHER OR NOT DIFFRACTION WAS PRESENT THE LAST TIME DIFPLA WAS CALLED FOR EDGE ME OF PLATE MP (ID=-1 INDICATES DIFFRACTION PRESENT) |
| IDD | DOUBLE DIFFRACTION SHADOW BOUNDARY IDENTIFICATION ARRAY |
| ISN | SIGN CHANGE VARIABLE |
| LHIT | SET TRUE IF RAY HITS A PLATE OR CYLINDER (FROM PLAINT OR CYLINT) |
| MC | CORNER AT END OF EDGE ME |
| ME | EDGE ON PLATE MP WHERE DIFFRACTION OCCURS |
| MP | PLATE FOR WHICH DIFFRACTION OCCURS |
| N | DO LOOP VARIABLE |
| PD | DOT PRODUCT OF EDGE BINORMAL AND DIFF RAY PROPAGATION DIR |

| | |
|---|---|
| PH | DIFFRACTED FIELD PHI POLARIZATION UNIT VECTOR (IN EDGE FIXED COORD SYS) IN RCS COMPONENTS |
| PHIR | PHI COMPONENT OF INCIDENT RAY PROPAGATION DIR IN RCS |
| PHO | INCIDENT FIELD PHI POLARIZATION UNIT VECTOR (IN EDGE FIXED COORD SYS) IN RCS COMPONENTS |
| PHSR | PHI COMPONENT OF DIF RAY PROPAGATION DIRECTION IN RCS |
| PP | NEGATIVE DOT PRODUCT OF EDGE BINORMAL AND INCIDENT RAY OPAGATION DIRECTION |
| PS | PSR*DPR |
| PSD | DIFFRACTED RAY PHI ANGLE IN EDGE-FIXED COORDINATE SYSTEM |
| PSO | PSOR*DPR |
| PSOD | INCIDENT RAY PHI ANGLE IN EDGE-FIXED COORDINATE SYSTEM |
| PSOR | PHI COMPONENT OF INCIDENT RAY DIRECTION IN EDGE FIXED COORDINATE SYSTEM |
| PSR | PHI COMPONENT OF DIFFRACTED RAY PROPAGATION DIRECTION IN EDGE-FIXED COORDIANTE SYSTEM |
| QD | DOT PRODUCT OF PLATE NORMAL AND DIFF RAY PROPAGATION DIR |
| QI | NEGATIVE OF DOT PRODUCT OF PLATE NORMAL AND INCIDENT RAY PROPAGATION DIRECTION |
| RM | MAGNITUDE OF VECTOR FROM CORNER MC TO SOURCE |
| RX<br>RY<br>RZ | X, Y, AND Z COMPONENTS OF VECTOR FROM CORNER MC TO SOURCE |
| SBO | SINE OF BO, THE ANGLE THE DIFFRACTED RAY MAKES WITH THE EDGE UNIT VECTOR |
| SNP | SINE OF HALF WEDGE ANGLE |
| SP | DISTANCE FROM SOURCE TO DIFFRACTION POINT (FROM SUB. DFPTND) |
| SPH | SINE OF PSR |
| SPHO | SINE OF PSOR |
| SPP | DISTANCE FROM SOURCE TO DIFFRACTION POINT |
| STHR | SINE OF THR |
| TERM | COEFFICIENT OF CORNER DIFFRACTED FIELDS |
| THIR | THETA COMPONENT OF INCIDENT RAY DIRECTION IN REF COORD SYS |
| THPR | ANGLE DIFFRACTED RAY MAKES WITH EDGE |
| THR | ANGLE BETWEEN EDGE UNIT VECTOR AND RAY FROM SOURCE TO CORNER MC |
| TPP | DISTANCE PARAMETER USED IN CALCULATING DIFFRACTION COEFFICIENTS |
| VECT | VECTOR USED TO MOVE DIFFRACTION POINT OFF EDGE FOR SHADOWING TESTS |
| VI | UNIT VECTOR OF INCIDENT RAY PROPAGATION DIR (FROM SUB. DFPTND) |
| VIP | UNIT VECTOR FROM SOURCE TO DIFFRACTION POINT |
| VMG | DISTANCE ALONG THE EDGE FROM FIRST CORNER OF EDGE TO DIFF POINT |
| VXS | 3X3 MATRIX DEFINING THE SOURCE COORDINATE SYSTEM AXES |
| XD | DIFFRACTION POINT (CALCULATED IN SUB. DFPTND) |
| XDP | DIFFRACTION POINT (USED FOR SHADOWING TESTS) |
| XS | SOURCE LOCATION IN REF COORD SYS |
| ZP | DOT PRODUCT OF DIFFRACTED RAY PROPAGATION DIRECTION UNIT VECTOR $\hat{D}$ AND VECTOR FROM DIF POINT TO CORNER MC |

CODE LISTING

```
C-----------------------------------------------------------------------
      SUBROUTINE DIFPLT(EDTH,EDPH,ECTH,ECPH,FNN,ME,MP)
C!!!
C!!!  DETERMINES THE DIFFRACTED FIELD FROM EDGE #ME ON PLATE #MP
C!!!  WITH THE PHASE REFERRED TO ORIGIN.
C!!!  SLOPE AND CORNER DIFF. IS OPTIONAL FROM INPUT DATA.
C!!!
      COMPLEX EF,EG,EIPR,EIPL,EXPH,EDPR,EDPL,EDTH,EDPH
      COMPLEX ECTH,ECPH,ECRI,ECRR,DS,DH,DPS,DPH
      COMPLEX EIPRP,EIPLP,EIX,EIY,EIZ,CORN,FFCT
      DIMENSION VI(3),XD(3),RHO(3),PH(3),BOP(3),RC(3),XDP(3),VIP(3)
      LOGICAL LSURF,LHIT,LSLOPE,LCORNR,LDIF,LDEBUG,LTEST
      COMMON/GEOPLA/X(14,6,3),V(14,6,3),VP(14,6,3),VN(14,3)
     2,MEP(14),MPX
      COMMON/EDMAG/VMAG(14,6)
      COMMON/SORINF/XS(3),VXS(3,3)
      COMMON/DIR/D(3),THSR,PHSR,SPHS,CPHS,STHS,CTHS
      COMMON/BNDFCL/BD(14,6,2)
      COMMON/THPHUV/DT(3),DP(2)
      COMMON/PIS/PI,TPI,DPR,RPD
      COMMON/LOGDIF/LSLOPE,LCORNR
      COMMON/TEST/LDEBUG,LTEST
      COMMON/DOUBLE/IDD(361),ID(14,6),IANG
      COMMON/HITPLT/MPH
      COMMON/SURFAC/LSURF(14)
      FN=FNN
C!!!  INITIALIZE FIELDS
      EDTH=(0.,0.)
      EDPH=(0.,0.)
      ECTH=(0.,0.)
      ECPH=(0.,0.)
      IF (LDEBUG) WRITE (6,100)
  100 FORMAT (/,' DEBUGGING DIFPLT SUBROUTINE')
C!!!  1.  PERFORM DIFFRACTION POINT GEOMETRY CALCULATIONS
      MC=ME+1
      IF(MC.GT.MEP(MP)) MC=1
      AFN=FN
      IF(FNN.GT.2.)AFN=6.-FNN
      CNP=COS(AFN*PI/2.)
      SNP=SIN(AFN*PI/2.)
      DV=0.
      DO 10 N=1,3
   10 DV=DV+D(N)*V(MP,ME,N)
      IF(ABS(DV).GT.0.999) GO TO 41
      BDEL=0.
      IF (LCORNR) BDEL=0.3
      BDLOW=BD(MP,ME,1)-BDEL
      BDHI=BD(MP,ME,2)+BDEL
C!!!  DETERMINE IF DIFFRACTION EXISTS
      IF(DV.LT.-BDLOW.OR.DV.GT.BDHI)GO TO 41
C!!!  COMPUTE EDGE DIFFRACTION POINT
      CALL DIFPNT(DS,DV,VI,SP,XD,[illegible])
      THIN=ATAN2(SQRT(VI(1)*VI(1)+VI(2)*VI(2)),VI(3))
      PHIN=ATAN2(VI(2),VI(1))
      ADN=0.
      VMC=0.
      DO 15 N=1,3
      XDP(N)=XD(N)
      VMC=VMC+(XD(N)-X(MP,ME,N))*V(MP,ME,N)
   15 ADN=ADN+(XS(N)-X(MP,1,N))*VN(MP,N)
      LDIF=.TRUE.
C!!!  IS DIFFRACTION POINT ON PLATE EDGE?
C!!!  IF NOT, SET DIFFRACTION POINT AT APPROPRIATE CORNER
C!!!  AND SET LDIF FALSE
      IF(VMC.LT.0.) GO TO [illegible]
      IF(VMG.LT.VMAG(MP,ME)-1.E-4)GO TO 1Ø2
      DO 1Ø3 N=1,3
  1Ø3 XDP(N)=X(MP,NC,N)-1.E-4*V(MP,ME,N)
      LDIF=.FALSE.
      GO TO 1Ø2
  1Ø1 DO 1Ø4 N=1,3
  1Ø4 XDP(N)=X(MP,ME,N)+1.E-4*V(MP,ME,N)
      LDIF=.FALSE.
  1Ø2 DO 16 N=1,3
      VECT=VP(MP,ME,N)*CNP+VN(MP,N)*SNP
   16 XDP(N)=XDP(N)+VECT*1.E-5
C!!!   2.  CHECK TO SEE IF RAY IS SHADOWED
C!!!   DETERMINE IF DIFFRACTED RAY HITS ANOTHER PLATE
       CALL PLAINT(XDP,D,DHIT,MP,LHIT)
       IF(LHIT) GO TO 42
C!!!   DETERMINE IF DIFFRACTED RAY HITS ANOTHER CYLINDER.
       CALL CYLINT(XDP,D,PHSR,DHIT,LHIT,.TRUE.)
       IF(LHIT) GO TO 44
       SPP=Ø.
       DO 111 N=1,3
       VIP(N)=XDP(N)-XS(N)
   111 SPP=SPP+VIP(N)*VIP(N)
       SPP=SQRT(SPP)
       DO 112 N=1,3
   112 VIP(N)=VIP(N)/SPP
C!!!   DOES RAY FROM SOURCE HIT ANOTHER PLATE OR A CYLINDER
C!!!   BEFORE DIFFRACTION?
       CALL PLAINT(XS,VIP,DHIT,MP,LHIT)
       IF(LHIT.AND.(DHIT.LT.SPP)) GO TO 42
       CALL CYLINT(XS,VIP,PHIR,DHIT,LHIT,.FALSE.)
       IF(LHIT.AND.(DHIT.LT.SPP)) GO TO 44
       IF (LDEBUG) WRITE (6,*) SP,VI,XD
       IF (LDEBUG) WRITE (6,*) SPP,VIP,XDP
C!!!   3.  CALCUATE DIFFRACTION ANGLES AND RELATED GEOMETRY
       QI=Ø.
       PP=Ø.
       QD=Ø.
       PD=Ø.
       DO 2Ø N=1,3
       QI=QI-VN(MP,N)*VI(N)
       PP=PP-VP(MP,ME,N)*VI(N)
       QD=QD+VN(MP,N)*D(N)
    2Ø PD=PD+VP(MP,ME,N)*D(N)
C!!!   CALCULATE PSO AND PS, THE INCIDENT AND DIFFRACTED PHI ANGLES
C!!!   IN EDGE-FIXED COORDINATE SYSTEM
       PSOR=BTAN2(QI,PP)
       PSO=DPR*PSOR
       IF(PSO.LT.Ø.) PSO=36Ø.+PSO
       PSR=BTAN2(QD,PD)
       PS=DPR*PSR
       IF(PS.LT.Ø.) PS=36Ø.+PS
       PSOD=PSO
       PSD=PS
       IF(FN.LE.2.)GO TO 21
       FN=FN-2.
       PSOD=36Ø.-PSO
       PSD=36Ø.-PS
    21 FNP=FN*18Ø.+1.E-4
       IF(PSOD.GT.FNP.OR.PSD.GT.FNP) GO TO 41
C!!!   IF RAY IS NOT SHADOWED, CHECK TO SEE IF DIFFRACTIONS JUST
C!!!   APPEARED.  IF SO SET FLAG IN ID(IANG)
       IF(ID(MP,ME).LE.-1)GO TO 22
       IDD(IANG)=-(4ØØ*ME+2Ø*MP+ID(MP,ME))
    22 ID(MP,ME)=-2
       SPHO=SIN(PSOR)
       CPHO=COS(PSOR)
      SPH=SIN(PSH)
      CPH=COS(PSH)
C!!!  COMPUTE DIFFRACTION POLARIZATION UNIT VECTORS(PHO,PH,BOP,BO)
      DO 30 N=1,3
      PHO(N)=-VP(MP,ME,N)*SPHO+VN(MP,N)*CPHO
30    PH(N)=-VP(MP,ME,N)*SPH+VN(MP,N)*CPH
      BOP(1)=PHO(2)*VI(3)-PHO(3)*VI(2)
      BOP(2)=PHO(3)*VI(1)-PHO(1)*VI(3)
      BOP(3)=PHO(1)*VI(2)-PHO(2)*VI(1)
      BO(1)=PH(2)*D(3)-PH(3)*D(2)
      BO(2)=PH(3)*D(1)-PH(1)*D(3)
      BO(3)=PH(1)*D(2)-PH(2)*D(1)
C!!!  COMPUTE SBO=SINE(BO)
      SBO=SQRT((V(MP,ME,3)*D(2)-V(MP,ME,2)*D(3))**2+(V(MP,ME,1)
     2*D(3)-V(MP,ME,3)*D(1))**2+(V(MP,ME,2)*D(1)-V(MP,ME,1)*D(2))
     2**2)
      TPP=SP*SBO*SBO
C!!!  4. CALCULATE EDGE DIFFRACTED FIELDS
C!!!  COMPUTE SOURCE PATTERN FACTORS
      CALL SOURCE(EF,EG,EIX,EIY,EIZ,THIR,PHIR,VXS)
      EIPR=EIX*PHO(1)+EIY*PHO(2)+EIZ*PHO(3)
      EIPL=EIX*BOP(1)+EIY*BOP(2)+EIZ*BOP(3)
C!!!  IF SLOPE DIFFRACTION IS DESIRED, COMPUTE INCIDENT SLOPE
C!!!  FIELD PATTERN FACTORS
      IF(LSLOPE)CALL SOURCP(EIPRP,EIPLP,VI,PHO,BOP,VXS)
C!!!  CALCULATE PHASE TERM (REFER PHASE TO RCS ORIGIN)
      GAM=XD(1)*D(1)+XD(2)*D(2)+XD(3)*D(3)
      EXPH=CEXP(CMPLX(0.,TPI*(GAM-SPS)))/SQRT(SP)
C!!!  COMPUTE EDGE DIFFRACTION COEFFICIENTS
      CALL DW (DS,DH,DPS,DPH,TPP,PSD,PSOD,SBO,FN,LSURF(MP))
      IF (LDEBUG) WRITE (6,*) EIPR,EIPL,EIPRP,EIPLP
      IF (LDEBUG) WRITE (6,*) DS,DH,DPS,DPH
      IF (LDEBUG) WRITE (6,*) TPP,PSD,PSOD,SBO,FN
C!!!  COMPUTE PERPENDICULAR AND PARALLEL COMPONENTS OF
C!!!  DIFF. FIELD(EDPR,EDPL)
      EDPR=-EIPR*DH*EXPH
      EDPL=-EIPL*DS*EXPH
      IF(.NOT.LSLOPE)GO TO 201
      EDPR=EDPR-EIPRP*DPH*EXPH/CMPLX(0.,TPI*SP*SBO)
      EDPL=EDPL-EIPLP*DPS*EXPH/CMPLX(0.,TPI*SP*SBO)
 201  IF (.NOT.LDIF) GO TO 202
C!!!  COMPUTE THETA AND PHI COMPONENTS OF EDGE DIFF. FIELD,
C!!!  IF DIFFRACTION EXISTS
      EDTH=EDPL*(BO(1)*DT(1)+BO(2)*DT(2)+BO(3)*DT(3))
     2+EDPR*(PH(1)*DT(1)+PH(2)*DT(2)+PH(3)*DT(3))
      EDPH=EDPL*(BO(1)*DP(1)+BO(2)*DP(2))
     2+EDPR*(PH(1)*DP(1)+PH(2)*DP(2))
C!!!  5.  IF CORNER DIFFRACTED FIELD IS DESIRED, CALCULATE
C!!!  CORNER FIELDS
 202  IF (.NOT.LCORNR) GO TO 40
      BETN=PSD-PSOD
      BETP=PSD+PSOD
      EF=(0.,0.)
      EG=(0.,0.)
      MC=ME-1
      ISN=1
C!!!  LOOP THRU BOTH CORNERS ON EDGE #ME.
35    MC=MC+1
      IF(MC.GT.MEP(MP)) MC=1
      ISN=-ISN
      RX=XS(1)-X(MP,MC,1)
      RY=XS(2)-X(MP,MC,2)
      RZ=XS(3)-X(MP,MC,3)
      RM=SQRT(RX*RX+RY*RY+RZ*RZ)
      CTH=V(MP,ME,1)*RX+V(MP,ME,2)*RY+V(MP,ME,3)*RZ
      CTH=ISN*CTH/RM
      CTHP=ISN*DV
      THPR=ACOS(CTHP)
      THR=ACOS(CTH)
      STHR=SIN(THR)
      DEL=2.*TPI*RM*(COS(.5*(THR+THPR))**2)
      ZP=(X(MP,MC,1)-XD(1))*D(1)+(X(MP,MC,2)-XD(2))*D(2)
     2+(X(MP,MC,3)-XD(3))*D(3)
      TERM=-STHR/TPI/(CTH+CTHP)/SQRT(RM)
C!!!  COMPUTE CORNER DIFFRACTION COEFFICIENT(CORN).
      CORN=-TERM*FFCT(DEL)*CEXP(CMPLX(Ø.,-TPI*(RM-SP-ZP)-.25*PI))
      CALL DI(ECBI,TPP,BETN,SBO,FN,DEL,.TRUE.)
      IF(LSURF(MP))GO TO 311
      CALL DI(ECBR,TPP,BETP,SBO,FN,DEL,.TRUE.)
C!!!  COMPUTE MODIFIED EDGE DIFF COEFFICIENTS(DH,DS).
      DH=ECBI+ECBR
      DS=ECBI-ECBR
      GO TO 312
  311 DH=ECBI
      DS=(Ø.,Ø.)
C!!!  COMPUTE CORNER DIFFRACTED FIELD COMPONENTS
C!!!  PARALLEL AND PERPENDICULAR TO EDGE
  312 EDPR=-EIPR*DH*EXPH
      EDPL=-EIPL*DS*EXPH
      IF(.NOT.LSLOPE)GO TO 2Ø3
      EDPR=EDPR-EIPRP*DPH*EXPH/CMPLX(Ø.,TPI*SP*SBO)
      EDPL=EDPL-EIPLP*DPS*EXPH/CMPLX(Ø.,TPI*SP*SBO)
C!!!  COMPUTE THETA AND PHI COMPONENTS OF CORNER
C!!!  DIFFRACTED FIELDS IN RCS
  2Ø3 ECTH=EDPL*(BO(1)*DT(1)+BO(2)*DT(2)+BO(3)*DT(3))
     2+EDPR*(PH(1)*DT(1)+PH(2)*DT(2)+PH(3)*DT(3))
      ECPH=EDPL*(BO(1)*DP(1)+BO(2)*DP(2))
     2+EDPR*(PH(1)*DP(1)+PH(2)*DP(2))
C!!!  COMPUTE TOTAL THETA AND PHI COMPONENTS OF CORNER
C!!!  DIFFRACTED FIELDS
      EF=EF+ECTH*CORN
      EG=EG+ECPH*CORN
      IF (.NOT.LDEBUG) GO TO 36
      WRITE (6,*) DS,DH,EDPR,EDPL
      WRITE (6,*) ECTH,ECPH,CORN
      WRITE (6,*) EF,EG
   36 CONTINUE
      IF(MC.EQ.ME) GO TO 35
      ECTH=EF
      ECPH=EG
      GO TO 4Ø
   41 ID(MP,ME)=-1
      GO TO 4Ø
C!!!  IF RAY IS SHADOWED,CHECK TO SEE IF DIFFRACTION
C!!!  JUST DISAPPEARED.  IF SO SET FLAG IN IDD
   44 MPH=Ø
   42 IF(ID(MP,ME).GE.-1)GO TO 43
      IDD(IANG)=-(4ØØ*ME+2Ø*MP+MPH)
   43 ID(MP,ME)=MPH
   4Ø IF (.NOT.LTEST) GO TO 2Ø4
      WRITE (6,2Ø5)
  2Ø5 FORMAT (/,' TESTING DIFPLT SUBROUTINE')
      WRITE (6,*) EDTH,EDPH,ECTH,ECPH
      WRITE (6,*) FN,ME,MP
  2Ø4 RETURN
      END
```

## DPI

### PURPOSE

To calculate the incident part or the reflection part of the wedge slope diffraction coefficient

### METHOD

This subroutine computes either the incident part or reflection part of the slope diffraction coefficient based on the uniform Geometrical Theory of Diffraction [10]. This coefficient is given as

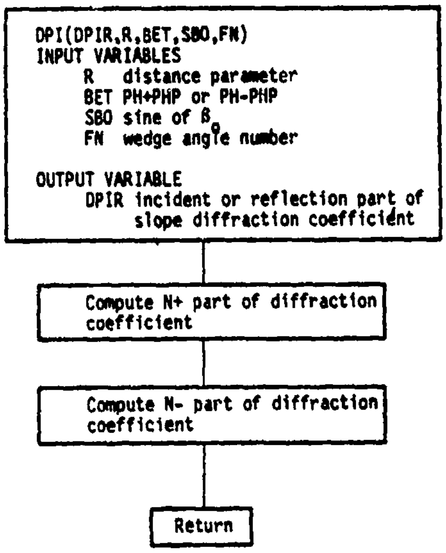
$$\mathrm{DPI}(R,\beta,\sin\beta_o,n) = \frac{-e^{-j\,\pi/4}}{4n^2\sqrt{2\pi k}\,\sin\beta_o}\left\{\csc^2\left(\frac{\pi+\beta}{2n}\right)F_s[kRa^+(\beta)] -\csc^2\left(\frac{\pi-\beta}{2n}\right)F_s[kRa^-(\beta)]\right\},$$

where

$$F_s(x) = 2jx[1-F(x)]$$

and where $\beta$, $a(\beta)$, $\bar{F}_s(x)$, $n$ are defined in the write up for subroutine DI. An illustration of the geometry is given in Figure 55.

## FLOW DIAGRAM


DPI(DPIR,R,BET,SBO,FN)
INPUT VARIABLES
R distance parameter
BET PH+PHP or PH-PHP
SBO sine of β$_o$
FN wedge angle number
OUTPUT VARIABLE
DPIR incident or reflection part of slope diffraction coefficient
Compute N+ part of diffraction coefficient
Compute N- part of diffraction coefficient
Return

## SYMBOL DICTIONARY

| | |
|---|---|
| A | ANGULAR FUNCTION FOR TRANSITION FUNCTION |
| ANG | BET IN RADIANS |
| BOTL | ARGUMENT OF TRANSITION FUNCTION |
| C | REAL PART OF FRESNEL INTEGRAL |
| COM | CONSTANT FOR SLOPE DIFFRACTION COEFFICIENT |
| CSCA | COSECANT TIMES THE A FUNCTION |
| DEN | 8*PI*FN*FN*SIN(BO) |
| DN | INTEGER WHICH MOST NEARLY SATISFIES THE EQUATION, 2*PI*FN*DN-BET=PI OR -PI |
| DNS | COMPUTATIONAL VARIABLE |
| EX | CEXP(J*K*R*A) |
| FPA | SLOPE TRANSITION FUNCTION WITHOUT THE A FUNCTION |
| N | COMPUTATIONAL VARIABLE |
| RAG | ARGUMENT OF COSECANT TERM |
| S | IMAGINARY PART OF FRESNEL INTEGRAL |
| SGN | SIGN OF DNS |
| TS | TSIN SQUARED |
| TSIN | SINE OF ARGUMENT OF COSECANT TERM |
| UNPI | N- COMPONENT OF DPI |
| UPPI | N+ COMPONENT OF DPI |

## CODE LISTING

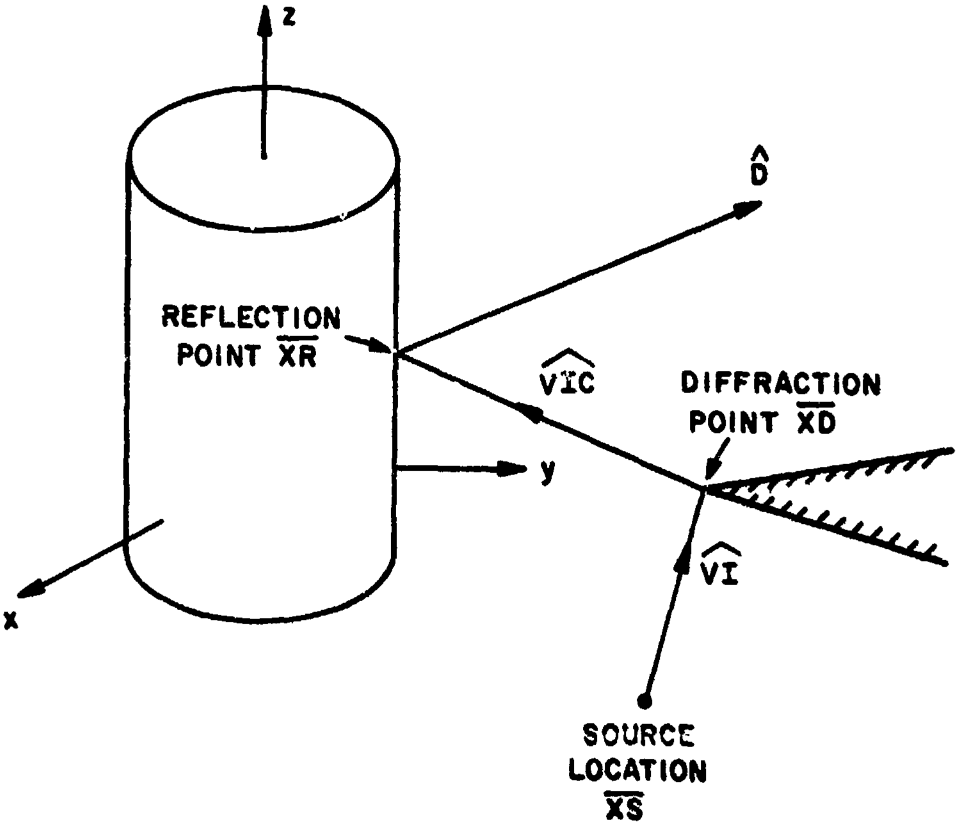
```
C--------------------------------------------------------------------
        SUBROUTINE DPI(DPIR,R,BET,SBO,FN)
C!!!
C!!!    INCIDENT (BET=PH-PHP) OR REFLECTED (BET=PH+PHP)
C!!!    PART OF WEDGE SLOPE DIFFRACTION COEFFICIENT
C!!!
        LOGICAL LDEBUG,LTEST
        COMMON/TEST/LDEBUG,LTEST
        COMPLEX TOP,COM,EX,UPPI,UNPI,FPA,DPIR
        COMMON/TOPD/TOP
        COMMON/PIS/PI,TPI,DPR,RPD
        IF (LDEBUG) WRITE (6,11)
   11   FORMAT (/,' DEBUGGING DPI SUBROUTINE')
        ANG=BET*RPD
        DEM=4.*TPI*FN*FN*SBO
        COM=-TOP/DEM
C!!!    N+ PART OF SLOPE DIFFRACTION COEFFICIENT
        DNS=(PI+ANG)/(2.0*FN*PI)
        SGN=SIGN(1.,DNS)
        N=IFIX(ABS(DNS)+0.5)
        DN=SGN*FLOAT(N)
        A=ABS(1.0+COS(ANG-2.0*FN*PI*DN))
        BOTL = 2.0*SQRT(ABS(R*A))
        EX=CEXP(CMPLX(0.0,TPI*R*A))
        CALL FRNELS (C,S,BOTL)
        C=SQRT(PI/2.0)*(0.5-C)
        S= SQRT(PI/2.0)*(S-0.5)
        FPA=TPI*R*(CMPLX(0.,2.)+4.*SQRT(ABS(TPI*R*A))*EX*CMPLX(C,S))
        RAG=(PI+ANG)/(2.0*FN)
        TSIN=SIN(RAG)
        TS=TSIN*TSIN
        IF(TS.GT.1.E-5) GO TO 442
        CSCA=-2.*FN*FN*COS(ANG-TPI*FN*DN)/COS((PI+ANG)/FN)
        GO TO 443
  442   CSCA=A/TS
  443   UPPI=COM*CSCA*FPA
        IF (LDEBUG) WRITE (6,*) DN,A,FPA,UPPI
C!!!    N- PART OF SLOPE DIFFRACTION COEFFICIENT
        DNS=(-PI+ANG)/(2.0*FN*PI)
        SGN=SIGN(1.,DNS)
        N=IFIX(ABS(DNS)+0.5)
        DN=SGN*FLOAT(N)
        A=ABS(1.0+COS(ANG-2.0*FN*PI*DN))
        BOTL = 2.0*SQRT(ABS(R*A))
        EX=CEXP(CMPLX(0.0,TPI*R*A))
        CALL FRNELS (C,S,BOTL)
        C=SQRT(PI/2.0)*(0.5-C)
        S= SQRT(PI/2.0)*(S-0.5)
        FPA=TPI*R*(CMPLX(0.,2.)+4.*SQRT(ABS(TPI*R*A))*EX*CMPLX(C,S))
        RAG=(PI-ANG)/(2.0*FN)
        TSIN=SIN(RAG)
        TS=TSIN*TSIN
        IF(TS.GT.1.E-5) GO TO 542
        CSCA=-2.*FN*FN*COS(ANG-TPI*FN*DN)/COS((PI-ANG)/FN)
        GO TO 123
  542   CSCA=A/TS
  123   UNPI=COM*CSCA*FPA
        IF (LDEBUG) WRITE (6,*) DN,A,FPA,UNPI
        DPIR=UPPI-UNPI
        IF (.NOT.LTEST) GO TO 2
        WRITE (6,1)
    1   FORMAT (/,' TESTING DPI SUBROUTINE')
        WRITE (6,*) DPIR,R,BET
        WRITE (6,*) SBO,FN
    2   RETURN
        END
```

DPLRCL

PURPOSE

To compute the far-zone electric field for a source ray diffracted off of a given edge on a given plate and then reflected by the cylinder.

PERTINENT GEOMETRY



z
$\hat{D}$
REFLECTION POINT $\overline{XR}$
$\widehat{VIC}$
DIFFRACTION POINT $\overline{XD}$
y
$\widehat{VI}$
x
SOURCE LOCATION $\overline{XS}$

Figure 57--Illustration of a ray diffracted off of a plate edge and then reflected by the cylinder.

METHOD

The field diffracted by a plate edge and then reflected by the elliptic cylinder is calculated in this subroutine. The field diffracted by a plate edge is found using the uniform Geometrical Theory of Diffraction[4]. This causes an astigmatic tube of rays to be incident on the cylinder. The field reflected by the cylinder is found using geometrical optics[4]. The resultant field in the far zone has the form (pp. 163-164, Reference 1)

$$\bar{E}^{d,r} = \bar{E}^{i}(Q_E) \cdot \bar{\bar{D}} \cdot \bar{\bar{R}} \sqrt{\frac{s'}{s''(s'+s'')}} \sqrt{\rho_1^r \rho_2^r}\, e^{-jks''} \frac{e^{-jks}}{s} \quad ,$$

where $\bar{E}^{i}(Q_E)$ is the incident field on the edge at $Q_E$, $\bar{\bar{D}}$ is the dyadic diffraction coefficient, $\bar{\bar{R}}$ is the dyadic reflection coefficient, $\rho_1^r$ and $\rho_2^r$ are the reflected ray caustic distances, s' is the distance from the source to the diffraction point, s" is the distance from the diffraction point to the reflection point, and s is the distance from the reflection point into the far zone. The geometry is shown in Figure 57, and further illustrations can be found in the write ups for subroutines REFCYL and DIFPLT. The phase of the field is referred to the reference coordinate system origin so that

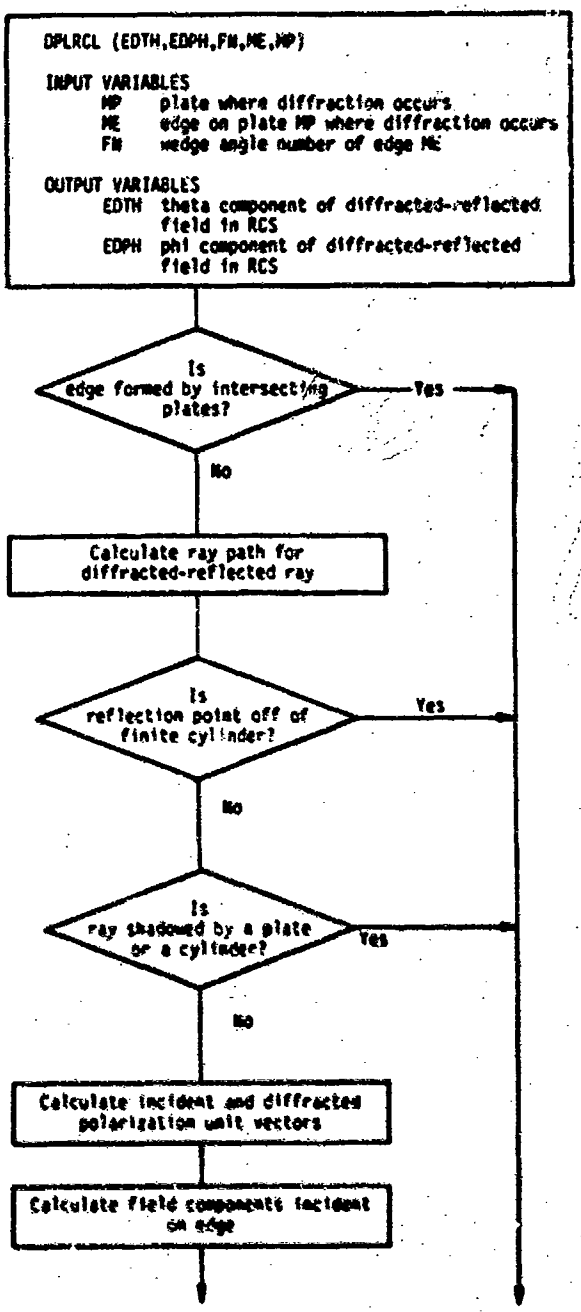
$$\frac{e^{-jks}}{s} = e^{jk\hat{D}\cdot\bar{X}_r} \frac{e^{-jkR}}{R} \quad .$$

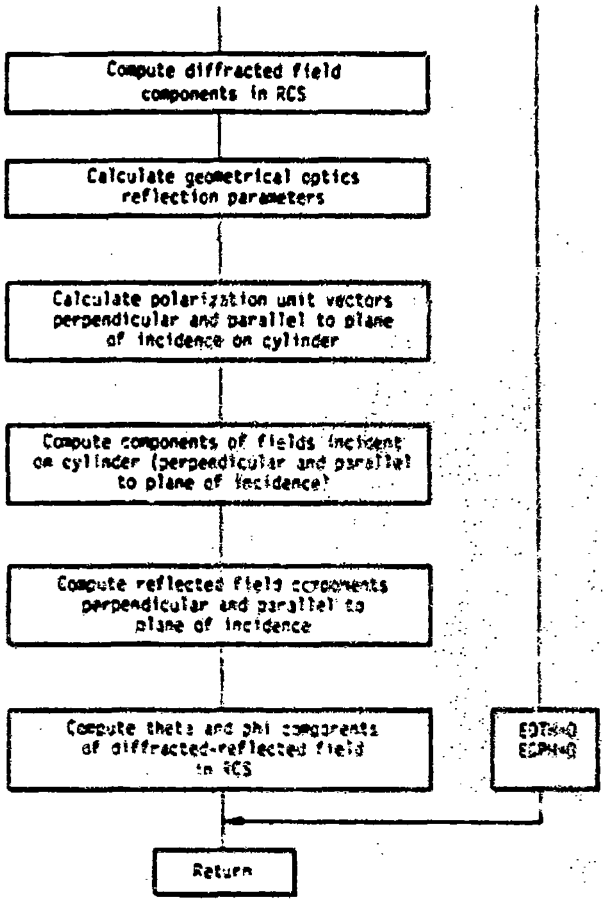
The diffracted-reflected field then has the form

$$\bar{E}^{d,r} = W_m(EDTH\hat{\theta}+EDPH\hat{\phi}) \frac{e^{-jkR}}{R} \quad ,$$

where the factor $\frac{e^{-jkR}}{R}$ and the source weight ($W_m$) are added elsewhere in the code.

FLOW DIAGRAM


DPLRCL (EDTH,EDPH,FN,ME,MP)
INPUT VARIABLES
MP plate where diffraction occurs
ME edge on plate MP where diffraction occurs
FN wedge angle number of edge ME
OUTPUT VARIABLES
EDTH theta component of diffracted-reflected field in RCS
EDPH phi component of diffracted-reflected field in RCS
Is edge formed by intersecting plates?
Yes
No
Calculate ray path for diffracted-reflected ray
Is reflection point off of finite cylinder?
Yes
No
Is ray shadowed by a plate or a cylinder?
Yes
No
Calculate incident and diffracted polarization unit vectors
Calculate field components incident on edge


Compute diffracted field components in RCS
Calculate geometrical optics reflection parameters
Calculate polarization unit vectors perpendicular and parallel to plane of incidence on cylinder
Compute components of fields incident on cylinder (perpendicular and parallel to plane of incidence)
Compute reflected field components perpendicular and parallel to plane of incidence
Compute theta and phi components of diffracted-reflected field in RCS
EDTH=0
EDPH=0
Return

## SYMBOL DICTIONARY

| | |
|---|---|
| BO | DIFFRACTED FIELD POLARIZATION UNIT VECTOR PARALLEL TO EDGE |
| BOP | INCIDENT FIELD POLARIZATION UNIT VECTOR PARALLEL TO EDGE |
| DD1 | DOT PRODUCT OF SOURCE RAY DIF FROM PLATE TANGENT TO TAN POINT 1 OF CYLINDER AND PROPAGATION DIRECTION (2-D) |
| DD2 | DOT PRODUCT OF SOURCE RAY DIF FROM PLATE TANGENT TO TAN POINT 2 OF CYLINDER AND PROPAGATION DIRECTION (2-D) |
| DH | DIFFRACTION COEF. FOR HARD BOUNDARY CONDITION |
| DHIT | DISTANCE TO HIT POINT ON PLATE |
| DOTP | TEST VARIABLE USED TO DETERMINE IF REFLECTION IS COMPUTED PROPERLY |
| DS | DIFFRACTION COEF. FOR SOFT BOUNDARY CONDITION |
| DV | DOT PRODUCT OF INCIDENT RAY PROPAGATION VECTOR AND EDGE UNIT VECTOR |
| EDPH | PHI COMPONENT OF EDGE DIFFRACTED REFLECTED E-FIELD |
| EDPL | COMPONENT OF DIFFRACTED FIELD PARALLEL TO THE EDGE |
| EDPR | COMPONENT OF DIFFRACTED FIELD PERPENDICULAR TO THE EDGE |
| EDTH | THETA COMPONENT OF EDGE DIFFRACTED REFLECTED E FIELD |
| EIPL | COMPONENT OF INCIDENT FIELD PARALLEL TO THE EDGE ON PLANE OF INCIDENCE |
| EIPR | COMPONENT OF INCIDENT FIELD PERPENDICULAR TO THE EDGE ON PLANE OF INCIDENCE |
| EIX<br>EIY<br>EIZ | SOURCE PATTERN FACTORS FOR X,Y, AND Z COMPENENTS OF INCIDENT E-FIELD |
| ERPP | COMPONENT OF REFLECTED E FIELD PARALLEL TO PLANE OF INCIDENCE |
| ERPR | COMPONENT OF REFLECTED E FIELD PERPENDICULAR TO PLANE OF INC. |
| ERX<br>ERY<br>ERZ | X,Y,Z COMPONENTS OF REFLECTED FIELD IN RCS |
| EXPH | COMPLEX PHASE AND SPREADING FACTOR |
| LDRC | SET TRUE IF STARTING POINT INFORMATION EXISTS FROM PREVIOUS PATTERN ANGLE |
| LHIT | SET TRUE IF PLATE IS HIT |
| ME | EDGE ON PLATE MP WHERE DIFFRACTION OCCURS |
| MP | PLATE FOR WHICH DIFFRACTION OCCURS |
| PD | DOT PRODUCT OF EDGE BINORMAL AND PROPAGATION DIRECTION |
| PH | DIFFRACTED FIELD PHI UNIT VECTOR PERPENDICULAR TO EDGE |
| PHIR | PHI COMPONENT OF PROPAGATION DIRECTION OF RAY INCIDENT ON PLATE MP |
| PHO | INCIDENT FIELD PHI UNIT VECTOR PERPENDICULAR TO EDGE |
| PP | NEGATIVE DOT PRODUCT OF EDGE BINORMAL AND INCIDENT RAY UNIT VECTOR |
| PS | DIFFRACTED RAY PHI ANGLE IN EDGE-FIXED COORDINATE SYSTEM |
| PSOR | PHI COMPONENT OF INCIDENT RAY DIRECTION IN EDGE FIXED COORDINATE SYSTEM |
| PSR | PHI COMPONENT OF DIF RAY PROPAGATION DIRECTION IN EDGE-FIXED COORD SYSTEM |
| QD | DOT PRODUCT OF PLATE NORMAL AND DIF RAY PROPAGATION DIRECTION |
| QI | NEGATIVE OF DOT PRODUCT OF PLATE NORMAL AND INCIDENT RAY UNIT VECTOR |
| RHI1 | RADIUS OF CURVATURE PERPENDICULAR TO EDGE OF DIFFRACTED RAY INCIDENT ON REFLECTION POINT |
| RHI2 | RADIUS OF CURVATURE IN EDGE PLANE OF DIFFRACTED RAY INCIDENT ON REFLECTION POINT |
| RHO1 | RAY SPREADING RADIUS IN PLANE OF CYLINDER CURVATURE AT REFLECTION POINT |
| RHO2 | RAY SPREADING RADIUS IN PLANE NORMAL TO PLANE OF INCIDENCE AT CYLINDER REFLECTION POINT |
| SRAD | DISTANCE FROM DIF POINT TO REFL POINT |
| SP | DISTANCE FROM SOURCE TO DIFFRACTION POINT (FROM SUB. DFRPPT) |
| THIR | THETA COMPONENT OF PROPAGATION DIRECTION OF RAY INCIDENT ON PLATE MP |
| UB | X,Y COMPONENTS OF UNIT VECTOR TANGENT TO CYL AT REFLECTION POINT |

```
UIPPX
UIPPY  } X,Y,Z COMPONENTS OF INCIDENT FIELD POLARIZATION UNIT VECTOR
UIPPZ    PARALLEL TO PLANE OF INCIDENCE
UIPRX
UIPRY  } X,Y,Z COMPONENTS OF INC/REF. FIELD POLARIZATION UNIT VECTOR
UIPRZ    PERPENDICULAR TO PLANE OF INCIDENCE
UN       X,Y COMPONENTS OF UNIT VECTOR NORMAL TO CYL AT REFLECTION
         POINT
URPPX
URPPY  } X,Y,Z COMPONENTS OF REFLECTED FIELD POLARIZATION UNIT VECTOR
URPPZ    PARALLEL TO PLANE OF INCIDENCE
VI       UNIT VECTOR OF PROP. DIR OF RAY INCIDENT AT DIFFRACTION
         POINT (FROM SUB. DFRPT)
VIC      X,Y,Z COMPONENTS OF UNIT VECTOR OF RAY DIRECTION
         BETWEEN DIFFRACTION AND REFLECTION
VR       ELLIPTICAL ANGLE DEFINING REFL. POINT ON CYL
         (2-D) IN RADS
VXS      3X3 MATRIX DEFINING THE SOURCE COORDINATE SYSTEM AXES
XD       X,Y,Z COMPONENTS OF DIFFRACTION POINT
XDP      MODIFIED DIFFRACTION POINT
XR       X,Y,Z COMPONENTS OF REFLECTION POINT
XS       SOURCE LOCATION IN REF COORD SYS
```

## CODE LISTING

```
C-------------------------------------------------------------------
      SUBROUTINE DPLRC (EDTH,EDPH,FN,ME,MP)
C!!!
C!!!  COMPUTES THE FIELD DIFFRACTED FROM EDGE #ME OF PLATE #MP
C!!!  THEN REFLECTED FROM THE ELLIPTIC CYLINDER
C!!!
      COMPLEX EF,EG,EIPR,EIPL,EXPH,DS,DH,DPS,DPH,EDPR,EDPL,EDTH,EDPH
      COMPLEX ERPR,ERPP,EIX,EIY,EIZ,ERX,ERY,ERZ
      DIMENSION UN(2),UB(2),VIC(3),XR(3)
      DIMENSION VI(3),XD(3),PHO(3),PH(3),BOP(3),BO(3),XDP(3)
      LOGICAL LHIT,LDRC,LDEBUG,LTEST
      COMMON/GEOPLA/X(14,6,3),V(14,6,3),VP(14,6,3),VN(14,3)
     2,MEP(14),MPX
      COMMON/SORINF/XS(3),VXS(3,3)
      COMMON/DIR/D(3),THSR,PHSR,SPHS,CPHS,STHS,CTHS
      COMMON/GEOMEL/A,B,ZC(2),SNC(2),CNC(2),CTC(2)
      COMMON/EDDFCI/BD(14,6,2)
      COMMON/EDDFCL/VDC(14,6),UDC(2),PDCR(14,6,2),TDCR(14,6,2)
     2,DTDC(14,6),BTDC(14,6,4),DDC(14,6,2)
      COMMON/INPHUV/DT(3),DP(2)
      COMMON/PIS/PI,TPI,DPR,RPD
      COMMON/TEST/LDEBUG,LTEST
      COMMON/CLDRC/LDRC(14,6)
C!!!  IS EDGE FORMED BY INTERSECTING PLATES?
      IF(FN.GT.2.) GO TO 40
C!!!  IS DIFFRACTION POSSIBLE?
      IF(D(3).GT.DDC(MP,ME,1).OR.D(3).LT.DDC(MP,ME,2)) GO TO 39
      DD1=BTDC(MP,ME,1)*CPHS+BTDC(MP,ME,2)*SPHS
      DD2=BTDC(MP,ME,3)*CPHS+BTDC(MP,ME,4)*SPHS
      IF(DD1.GT.DTDC(MP,ME).AND.DD2.GT.DTDC(MP,ME)) GO TO 39
C!!!  CALCULATE RAY PATH FOR DIFFRACTED-REFLECTED FIELD
      CALL DFRFPT(VR,XR,DOTP,DD,SHAG,VIC,XD,SP,VI,DV,ME,MP
     2,LDRC(MP,ME))
      IF(DOTP.LE.0.) GO TO 40
      IF(DV.LT.BD(MP,ME,1).OR.DV.GT.BD(MP,ME,2)) GO TO 40
C!!!  IS REFLECTION OF POINT OFF OF FINITE CYLINDER?
      IF(XR(3).GT.ZC(1)+XR(1)*CTC(1).OR.
     2XR(3).LT.ZC(2)+XR(1)*CTC(2)) GO TO 40
      CNP=COS(FN*0.5*PI)
      SNP=SIN(FN*0.5*PI)
      DO 16 N=1,3
      VECT=VP(MP,ME,N)*CNP+VN(MP,N)*SNP
   16 XDP(N)=XD(N)+VECT*1.E-5
C!!!  IS RAY SHADOWED BY A PLATE OR A CYLINDER?
      CALL PLAINT(XDP,VIC,DHIT,MP,LHIT)
      IF(LHIT.AND.(DHIT.LT.SHAG)) GO TO 40
      CALL PLAINT(XS,VI,DHIT,MP,LHIT)
      IF(LHIT.AND.(DHIT.LT.SP)) GO TO 40
      CALL PLAINT(XR,0,DHIT,0,LHIT)
      IF(LHIT) GO TO 40
      THIR=ATAN2(SQRT(VI(1)*VI(1)+VI(2)*VI(2)),VI(3))
      PHIR=ATAN2(VI(2),VI(1))
      CALL CYLINT(XS,VI,PHIR,DHIT,LHIT,.FALSE.)
      IF(LHIT.AND.(DHIT.LT.SP)) GO TO 40
      CI=0.
      PP=0.
      QQ=0.
      PQ=0.
      DO 20 N=1,3
      CI=CI-VN(MP,N)*VI(N)
      PP=PP-VP(MP,ME,N)*VI(N)
      QQ=QQ+VN(MP,N)*VIC(N)
      PQ=PQ+VP(MP,ME,N)*VIC(N)
      PSO=ATAN2(CI,PP)
      PSO=RPD*PSO
      IF(PSO.LT.0.) PSO=360.+PSO
      PSR=ATAN2(QD,PD)
      PS=DPR*PSR
      IF(PS.LT.0.) PS=360.+PS
      FNP=FN*180.+1.E-4
      IF(PSO.GT.FNP.OR.PS.GT.FNP) GO TO 40
      SPHO=SIN(PSOR)
      CPHO=COS(PSOR)
      SPH=SIN(PSR)
      CPH=COS(PSR)
C!!!  CALCULATE INCIDENT AND DIFFRACTED POLARIZATION
C!!!  UNIT VECTORS
      DO 30 N=1,3
      PHO(N)=-VP(MP,ME,N)*SPHO+VN(MP,N)*CPHO
   30 PH(N)=-VP(MP,ME,N)*SPH+VN(MP,N)*CPH
      BOP(1)=PHO(2)*VI(3)-PHO(3)*VI(2)
      BOP(2)=PHO(3)*VI(1)-PHO(1)*VI(3)
      BOP(3)=PHO(1)*VI(2)-PHO(2)*VI(1)
      BO(1)=PH(2)*VIC(3)-PH(3)*VIC(2)
      BO(2)=PH(3)*VIC(1)-PH(1)*VIC(3)
      BO(3)=PH(1)*VIC(2)-PH(2)*VIC(1)
C!!!  CALCULATE SOURCE FIELD PATTERN FACTOR
      CALL SOURCE(EF,EG,EIX,EIY,EIZ,THIR,PHIR,VXS)
      EIPR=EIX*PHO(1)+EIY*PHO(2)+EIZ*PHO(3)
      EIPL=EIX*BOP(1)+EIY*BOP(2)+EIZ*BOP(3)
      SBO=SQRT((V(MP,ME,3)*VIC(2)-V(MP,ME,2)*VIC(3))**2
     2+(V(MP,ME,1)*VIC(3)-V(MP,ME,3)*VIC(1))**2
     2+(V(MP,ME,2)*VIC(1)-V(MP,ME,1)*VIC(2))**2)
      TPP=SP*SMAG*SBO*SBO/(SP+SMAG)
      EXPH=CEXP(CMPLX(0.,-TPI*SP))/SQRT(SP)
C!!!  CALCULATE DIFFRACTED FIELDS
      CALL DN(DS,DH,DPS,DPH,TPP,PS,PSO,SBO,FN,.FALSE.)
      EDPR=-EIPR*DH*EXPH
      EDPL=-EIPL*DS*EXPH
C!!!  CALCULATE GEOMETRICAL OPTICS REFLECTION
C!!!  PARAMETERS
      RG=DD*DD*DD/A/B
      CALL NANDR(UI,UB,VR)
      CTHC=UN(1)*D(1)+UN(2)*D(2)
      WR=ATAN2(-VIC(1)*UB(1)-VIC(2)*UB(2),-VIC(3))
      RHI1=SMAG
      RHI2=SMAG+SP
      TH11=PH(1)*UB(1)+PH(2)*UB(2)
      TH12=PH(3)
      TH21=BO(1)*UB(1)+BO(2)*UB(2)
      TH22=BO(3)
      DET=TH11*TH22-TH12*TH21
      CTHD=CTHC/(DET*DET)
      RHA=.5*(1./RHI1+1./RHI2)+CTHD*(TH22*TH22+TH12*TH12)/
      RH12=1./RHI1-1./RHI2
      RHB=RH12*RH12+RH12*4.*CTHD*(TH22*TH22-TH12*TH12)/RG
      RHB=RHB+4.*CTHD*CTHD*((TH22*TH22+TH12*TH12)/RG)**2
      RHB=.5*SQRT(RHB)
      RHO1=1./(RHA+RHB)
      RHO2=1./(RHA-RHB)
C!!!  COMPUTE POLARIZATION UNIT VECTORS (PERPENDICULAR
C!!!  AND PARALLEL TO PLANE OF INCIDENCE)
      UIPRX=SIN(WR-.5*PI)*UB(1)
      UIPRY=SIN(WR-.5*PI)*UB(2)
      UIPRZ=COS(WR-.5*PI)
      UIPPX=VIC(3)*UIPRY-VIC(2)*UIPRZ
      UIPPY=VIC(1)*UIPRZ-VIC(3)*UIPRX
      UIPPZ=VIC(2)*UIPRX-VIC(1)*UIPRY
      URPPX=D(3)*UIPRY-D(2)*UIPRZ
      URPPY=D(1)*UIPRZ-D(3)*UIPRX
      URPPZ=D(2)*UIPRX-D(1)*UIPRY
      EXPH=CEXP(CMPLX(0.,-TPI*SMAG))/SQRT(SMAG*(SP+SMAG))
```

```
C!!!  CALCUALTE DIFFRACTED FIELD COMPONENTS INCIDENT
C!!!  ON CYLINDER PARALLEL AND PERP. TO PLANE OF INC.
      EIPR=EDPL*(BO(1)*UIPRX+BO(2)*UIPRY+BO(3)*UIPRZ)
     2+EDPR*(PH(1)*UIPRX+PH(2)*UIPRY+PH(3)*UIPRZ)
      EIPL=EDPL*(BO(1)*UIPPX+BO(2)*UIPPY+BO(3)*UIPPZ)
     2+EDPR*(PH(1)*UIPPX+PH(2)*UIPPY+PH(3)*UIPPZ)
C!!!  COMPUTE REFLECTED FIELD COMPONENTS PARALLEL
C!!!  AND PERPENDICULAR TO CYLINDER
      ERPR=-SQRT(RHO1*RHO2)*EXPH*EIPR
      ERPP=SQRT(RHO1*RHO2)*EXPH*EIPL
C!!!  CALCULATE X,Y,Z COMPONENTS OF REFLECTED FIELD
      ERX=ERPR*UIPRX+ERPP*URPPX
      ERY=ERPR*UIPRY+ERPP*URPPY
      ERZ=ERPR*UIPRZ+ERPP*URPPZ
      EXPH=CEXP(CMPLX(Ø.,TPI*(XR(1)*D(1)+XR(2)*D(2)+XR(3)*D(3))))
C!!!  COMPUTE THETA AND PHI COMPONENTS OF DIFFRACTED-
C!!!  REFLECTED FIELD IN RCS
      EDTH=(ERX*DT(1)+ERY*DT(2)+ERZ*DT(3))*EXPH
      EDPH=(ERX*DP(1)+ERY*DP(2))*EXPH
      GO TO 9ØØ
   39 LDRC(MP,ME)=.FALSE.
   4Ø CONTINUE
      EDTH=(Ø.,Ø.)
      EDPH=(Ø.,Ø.)
  9ØØ CONTINUE
      IF(.NOT.LTEST) RETURN
      WRITE(6,9Ø1)
  9Ø1 FORMAT(/,' TESTING DPLRCL SUBROUTINE')
      WRITE(6,*) EDTH,EDPH,FN,ME,MP
      RETURN
      END
```

## DPLRPL

PURPOSE

To calculate the far-zone electric field (with phase referred to the RCS origin) for a source ray which diffracts off of edge ME of plate MP and is then reflected by plate MR.

PERTINENT GEOMETRY

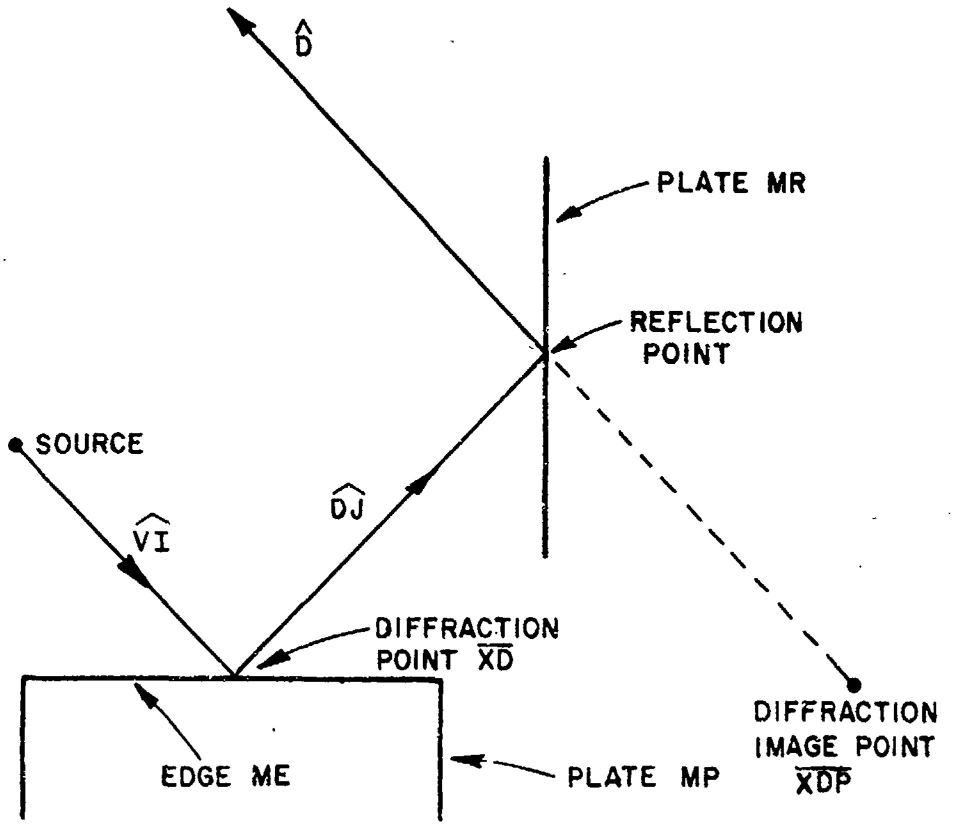


Figure 58--Illustration of edge-diffracted, plate-reflected ray.

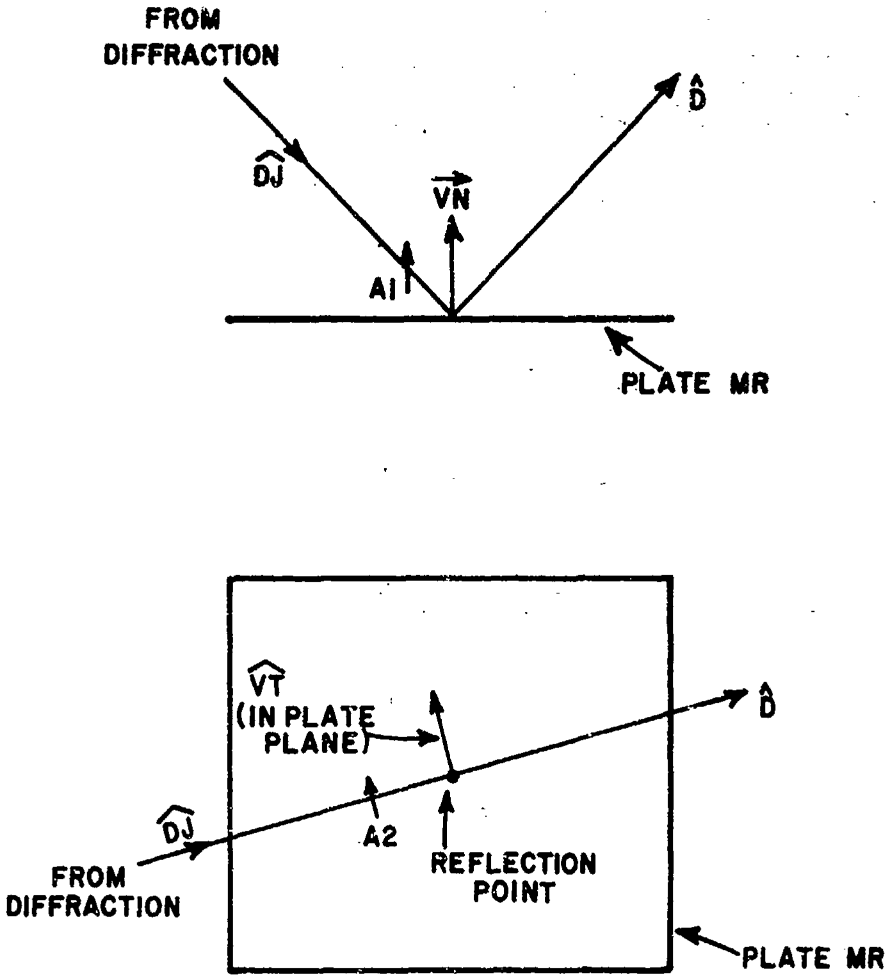


Figure 59--Geometry used in computing plate reflection.

METHOD

The fields diffracted by a plate edge and the reflected by another plate are calculated in this subroutine[4,9,10]. The diffracted and slope diffracted fields of the plate edges and corners are obtained as described in subroutine DIFPLT. The reflection from the plate is found by decomposing the diffracted fields into components tangent and normal to the reflection plate (see Figure 59), satisfying the appropriate boundary conditions and then transforming the field back to the reference coordinate system. The edge and slope diffracted fields are combined and the phase referred to the reference coordinate system origin by the factor $e^{jk\hat{D}\cdot\overline{XDP}}$. The form of the

the field is therefore given by

$$\overline{E}^d = W_m(EDTH\hat{\theta} + EDPH\hat{\phi})\,\frac{e^{-jkR}}{R} \cdot$$

The corner diffracted and slope corner diffracted fields are combined in a similar way and are given by

$$\overline{E}^c = W_m(ECTH\hat{\theta} + ECPH\hat{\phi})\,\frac{e^{-jkR}}{R} \quad ,$$

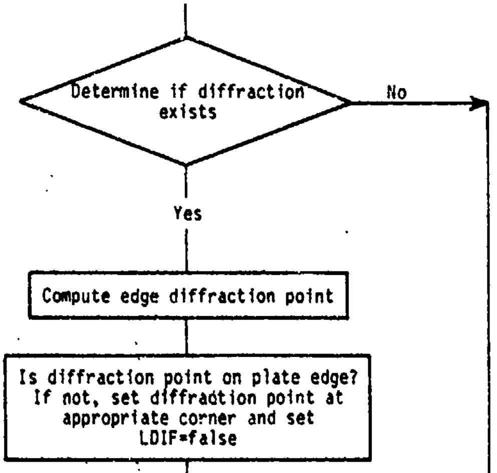
where the factor $\frac{e^{-jkR}}{R}$ and the source weight ($W_m$) are added elsewhere in the code.

## FLOW DIAGRAM

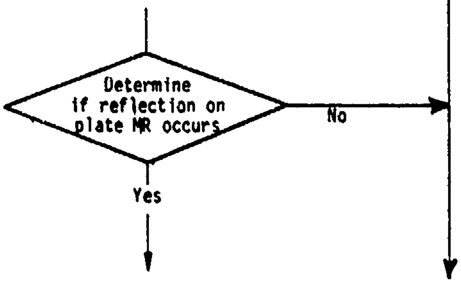
```
DPLRPL (EDTH,EDPH,ECTH,ECPH,FNN,ME,MP,MR)

INPUT VARIABLES
    FNN   wedge angle indicator
    MP    plate where diffraction occurs
    ME    edge on plate MP where diffraction occurs
    MR    plate where reflection occurs

OUTPUT VARIABLES
    EDTH  theta component of edge diffracted,
          reflected E field in RCS
    EDPH  phi component of edge diffracted,
          reflected E field in RCS
    ECTH  theta component of corner diffracted,
          reflected E field in RCS
    ECPH  phi component of corner diffracted,
          reflected E field in RCS
```

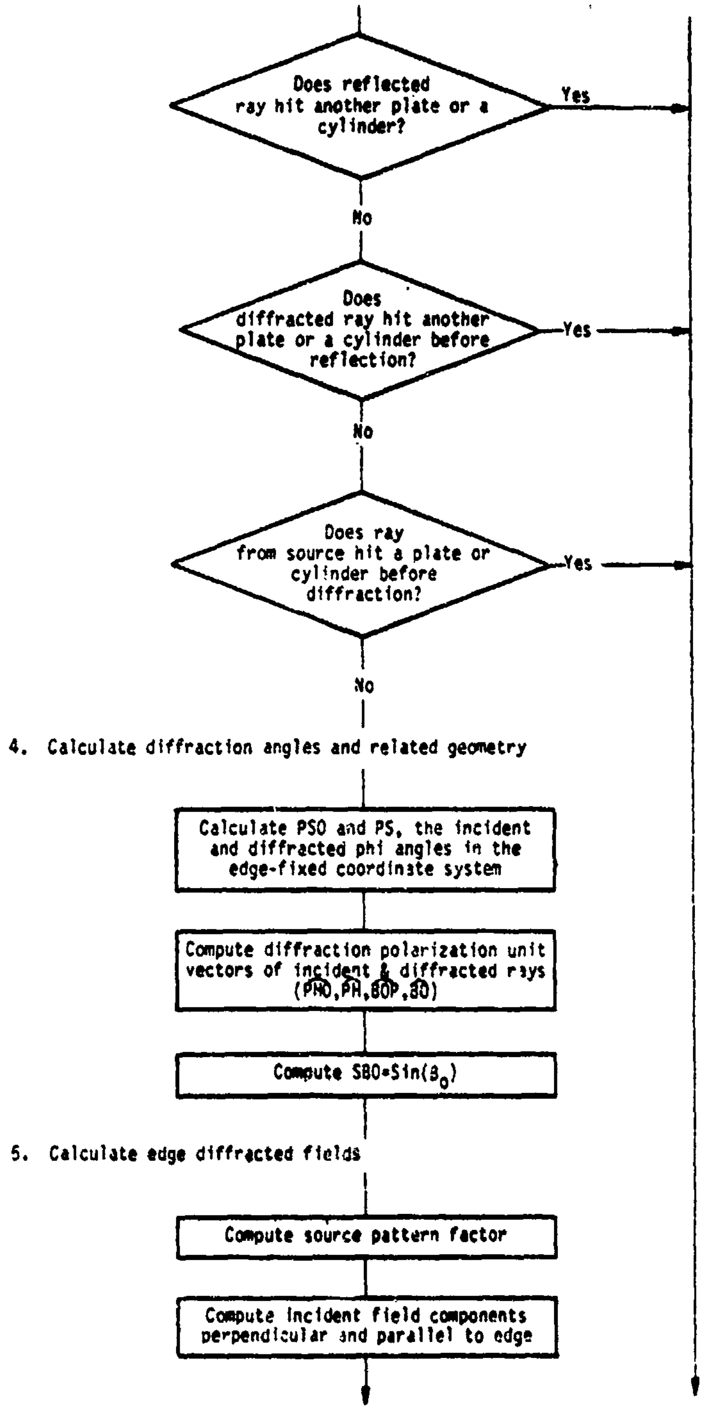
1. Compute direction $\hat{DJ}$ of ray incident on plate MR (ray propagation direction after diffraction).

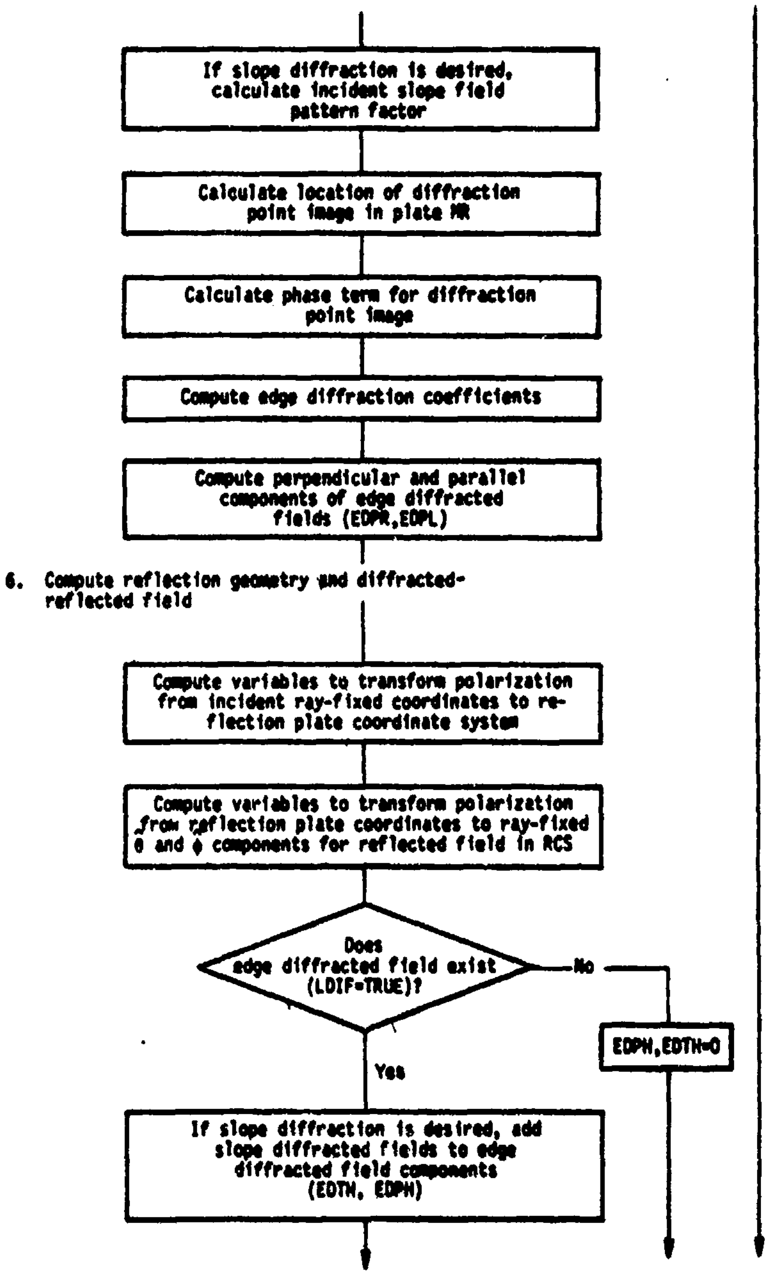
2. Perform diffraction point geometry calculations to obtain diffracted ray in direction $\hat{DJ}$

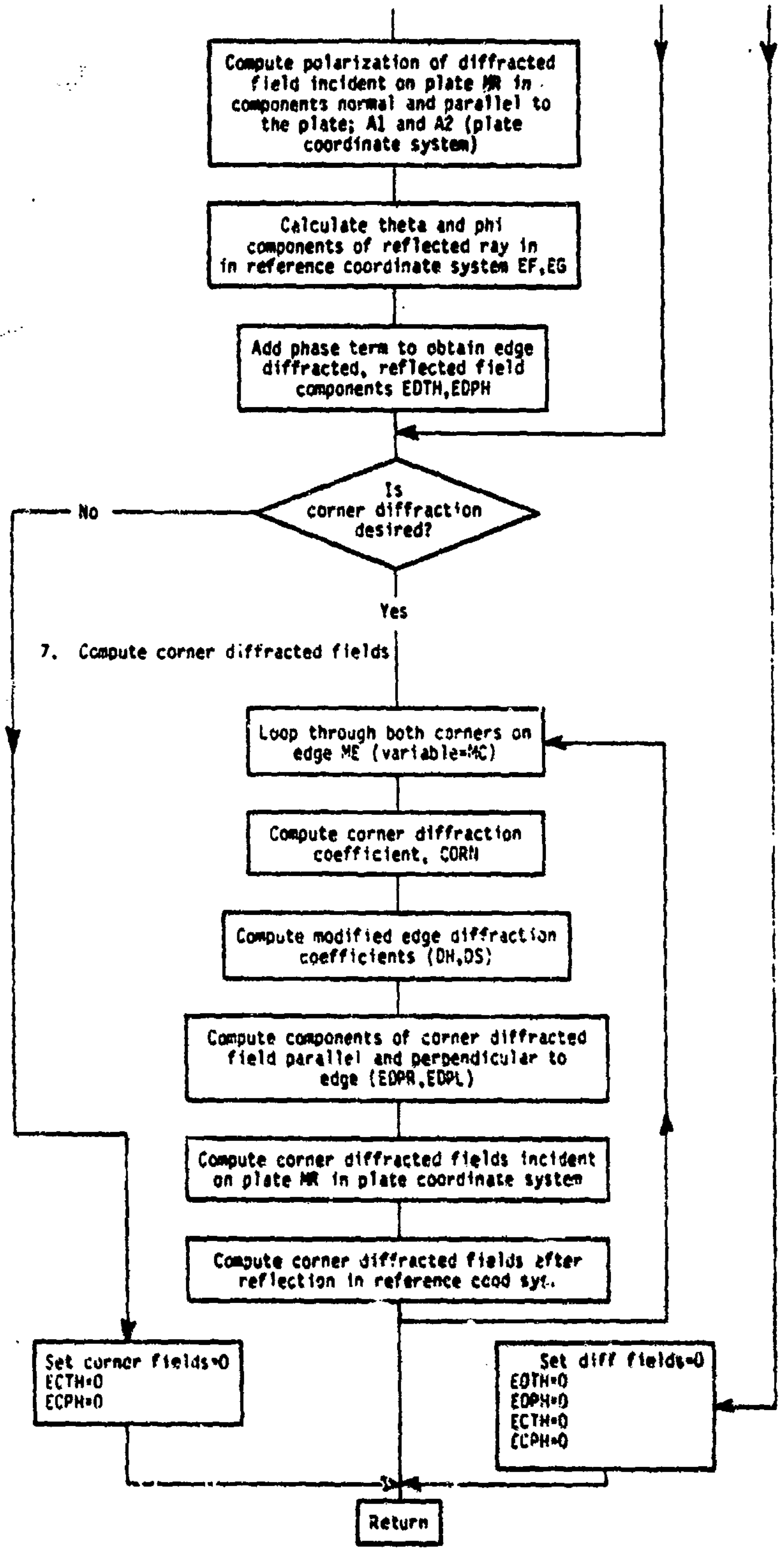

Determine if diffraction exists
No
Yes
Compute edge diffraction point
Is diffraction point on plate edge? If not, set diffradtion point at appropriate corner and set LDIF=false

3. Check to see if ray is shadowed anywhere




Does reflected ray hit another plate or a cylinder?
Yes
No
Does diffracted ray hit another plate or a cylinder before reflection?
Yes
No
Does ray from source hit a plate or cylinder before diffraction?
Yes
No
4. Calculate diffraction angles and related geometry
Calculate PSO and PS, the incident and diffracted phi angles in the edge-fixed coordinate system
Compute diffraction polarization unit vectors of incident & diffracted rays (PHO,PH,BOP,BO)
Compute SBO=Sin(β0)
5. Calculate edge diffracted fields
Compute source pattern factor
Compute incident field components perpendicular and parallel to edge


If slope diffraction is desired, calculate incident slope field pattern factor
Calculate location of diffraction point image in plate MR
Calculate phase term for diffraction point image
Compute edge diffraction coefficients
Compute perpendicular and parallel components of edge diffracted fields (EDPR,EDPL)
6. Compute reflection geometry and diffracted-reflected field
Compute variables to transform polarization from incident ray-fixed coordinates to reflection plate coordinate system
Compute variables to transform polarization from reflection plate coordinates to ray-fixed θ and ϕ components for reflected field in RCS
Does edge diffracted field exist (LDIF=TRUE)?
No
EDPH,EDTH=0
Yes
If slope diffraction is desired, add slope diffracted fields to edge diffracted field components (EDTH, EDPH)


Compute polarization of diffracted field incident on plate MR in components normal and parallel to the plate; A1 and A2 (plate coordinate system)
Calculate theta and phi components of reflected ray in in reference coordinate system EF,EG
Add phase term to obtain edge diffracted, reflected field components EDTH,EDPH
Is corner diffraction desired?
No
Yes
7. Compute corner diffracted fields
Loop through both corners on edge ME (variable=MC)
Compute corner diffraction coefficient, CORN
Compute modified edge diffraction coefficients (DH,DS)
Compute components of corner diffracted field parallel and perpendicular to edge (EDPR,EDPL)
Compute corner diffracted fields incident on plate MR in plate coordinate system
Compute corner diffracted fields after reflection in reference ccod syt.
Set corner fields=0
ECTH=0
ECPH=0
Set diff fields=0
EDTH=0
EDPH=0
ECTH=0
ECPH=0
Return

## SYMBOL DICTIONARY

| | |
|---|---|
| A1 | COMPONENT OF INCIDENT DIF FIELD NORMAL TO PLATE MR |
| A2 | COMPONENT OF INCIDENT DIF FIELD TANGENT TO PLATE MR |
| A3 | DETERMINANT OF TRANSFORMATION MATRIX |
| ADN | DOT PRODUCT OF VECTOR FROM PLATE MP TO THE SOURCE AND THE PLATE UNIT NORMAL |
| AFN | WEDGE ANGLE NUMBER |
| BDEL | VARIABLE USED TO EXPAND DIFFRACTION ANGLE RANGE IF CORNER DIFFRACTION IS USED |
| BDHI | UPPER LIMIT FOR BD, THE COSINE OF THE DIFFRACTION ANGLE BETA |
| BDLOW | LOWER LIMIT FOR BD, THE COSINE OF THE DIFFRACTION ANGLE BETA |
| BETN | DIFFERENCE IN DIFFRACTED AND INCIDENT PHI ANGLES |
| BETP | SUM OF DIFFRACTED AND INCIDENT PHI ANGLES |
| BO | DIFFRACTED FIELD BETA POLARIZATION UNIT VECTOR (IN EDGE FIXED COORD SYS) IN RCS COMPONENTS (FOR DIF EDGE)<br>IN (X,Y,Z) REF COORD SYS. COMPONENTS |
| BOP | INCIDENT FIELD BETA POLARIZATION UNIT VECTOR (IN EDGE FIXED COORD SYS) IN RCS COMPONENTS (FOR DIF EDGE) |
| C11 | DOT PRODUCT OF REFLECTED FIELD POLARIZATION VECTOR DT AND PLATE COORD SYS UNIT VECTOR VN |
| C11A | DOT PRODUCT OF RAY-FIXED C.S. VECTOR BO AND PLATE C.S. VECTOR VN |
| C12 | DOT PRODUCT OF RAY FIXED COORD SYS VECTOR DP AND PLATE COORD SYS UNIT VECTOR VN |
| C12A | DOT PRODUCT OF RAY-FIXED C.S. VECTOR PH AND PLATE C.S. VECTOR VT |
| C21 | DOT PRODUCT OF RAY FIXED COORD SYS VECTOR DT AND PLATE COORD SYS UNIT VECTOR VT |
| C21A | DOT PRODUCT OF RAY FIXED COORD SYS VECTOR BO AND PLATE COORD SYS VECTOR VT |
| C22 | DOT PROD. OF REFLECTED FIELD POLARIZATION UNIT VECTOR DP AND PLATE COORD SYS UNIT VECTOR VT<br>COORD SYS UNIT VECTOR VT |
| C22A | DOT PRODUCT OF RAY-FIXED C.S. VECTOR PH AND PLATE C.S. VECTOR VT |
| CNP | COSINE OF HALF WEDGE ANGLE |
| CORN | CORNER DIFFRACTION COEFFICIENT |
| CPH | COSINE OF PSR |
| CPHJ | COSINE OF PHJR |
| CPHO | COSINE OF PSOR |
| CTH | COSINE OF THR |
| CTHJ | COSINE OF THJR |
| CTHP | COSINE OF THPR |
| DEL | PARAMETER USED IN TRANSITION FUNCTION |
| DH | DIFFRACTION COEF. FOR HARD BOUNDARY CONDITION |
| DHIT | DISTANCE FROM SOURCE TO NEAREST HIT (FROM SUBS. PLAINT OR CYLINT) |
| DHT | DISTANCE FROM SOURCE TO HIT (RETURNED FROM PLAINT AND CYLINT) |
| DJ | X,Y, AND Z COMPONENTS OF RAY PROP. DIRECTION BETWEEN DIFFRACTION AND REFLECTION |
| DPH | SLOPE DIFFRACTION COEFFICIENT FOR HARD BOUNDARY CONDITION |
| DPS | SLOPE DIFFRACTION COEFFICIENT FOR SOFT BOUNDARY CONDITION |
| DS | DIFFRACTION COEF. FOR SOFT BOUNDARY CONDITION |
| DV | DOT PRODUCT OF EDGE VECTOR AND DIFFRACTED RAY PROPAGATION DIRECTION UNIT VECTOR, DJ |
| ECBI | DIFFRACTION COEFFICIENT (FROM SUB. DI) FOR INCIDENT DIFFRACTED FIELD, MODIFIED FOR CORNER DIFFRACTION |
| ECBR | EDGE DIFFRACTION COEFFICIENT (FROM SUB. DI) FOR REFLECTED DIFFARACTED FIELD, MODIFIED FOR CORNER DIFFRACTION |
| ECPH | PHI COMPONENT OF CORNER DIFFRACTED, REFLECTED E-FIELD |
| ECTH | THETA COMPONENT OF CORNER DIFFRACTED, REFLECTED E-FIELD |
| EDPH | PHI COMPONENT OF EDGE DIFFRACTED, REFLECTED E-FIELD |
| EDPL | COMPONENT OF DIFFRACTED FIELD PARALLEL TO THE EDGE |
| EDPR | COMPONENT OF DIFFRACTED FIELD PERPENDICULAR TO THE EDGE |
| EDTH | THETA COMPONENT OF EDGE DIFFRACTED, REFLECTED E-FIELD |
| EF | THETA COMPONENT OF PATTERN FACTOR OF FIELD INCIDENT ON EDGE ALSO THETA COMPONENT OF REFLECTED FIELD |
| EG | PHI COMPONENT OF PATTERN FACTOR OF FIELD INCIDENT ON EDGE ALSO PHI COMPONENT OFN REFLECTED FIELD IN RCS |
| EIPL | COMPONENT OF INCIDENT FIELD PARALLEL TO THE EDGE |

| | |
|---|---|
| EIPLP | PATTERN FACTOR FOR COMPONENT OF SOURCE (INCIDENT) SLOPE FIELD PARALLEL TO THE EDGE (RAY INCIDENT ON DIFF EDGE) |
| EIPR | COMPONENT OF INCIDENT FIELD PERPENDICULAR TO THE EDGE |
| EIPRP | PATTERN FACTOR FOR COMPONENT OF SOURCE (INCIDENT) SLOPE FIELD PERPENDICULAR TO THE EDGE (RAY INCIDENT ON DIFF EDGE) |
| EIX<br>EIY<br>EIZ | SOURCE PATTERN FACTORS FOR X,Y, AND Z COMPENENTS OF INCIDENT FIELD ON EDGE |
| EXPH | COMPLEX PHASE TERM (REFER PHASE TO RCS. ORIGIN) |
| FN | WEDGE ANGLE NUMBER |
| FNN | WEDGE ANGLE INDICATOR |
| FNP | ANGLE EXTERIOR TO WEDGE IN DEGREES |
| GAM | DOT PRODUCT OF THE PROPAGATION DIRECTION AND THE VECTOR FROM THE REF COORD SYS ORIGIN TO THE DIFFRACTION POINT IMAGE LOCATION |
| ISN | SIGN CHANGE VARIABLE |
| LHIT | SET TRUE IF RAY HITS A PLATE OR CYLINDER (FROM PLAINT OR CYLINT) |
| MC | CORNER AT END OF EDGE ME |
| ME | EDGE ON PLATE MP WHERE DIFFRACTION OCCURS |
| MP | PLATE FOR WHICH DIFFRACTION OCCURS |
| MR | PLATE WHERE REFLECTION OCCURS |
| N | DO LOOP VARIABLE |
| PD | DOT PRODUCT OF EDGE BINORMAL AND PROPAGATION DIRECTION |
| PH | DIFFRACTED FIELD PHI POLARIZATION UNIT VECTOR (IN EDGE FIXED COORDINATE SYSTEM)IN RCS COMPONENTS (FOR DIF EDGE) |
| PHIR | PHI COMPONENT OF INCIDENT RAY DIRECTION IN REF COORD SYS. |
| PHJR | PHI COMPONENT OF RAY PROP. DIR. BETWEEN DIF AND REFLECTION IN RCS |
| PHO | INCIDENT FIELD PHI POLARIZATION UNIT VECTOR (IN EDGE FIXED COORD SYS) IN RCS COMPONENTS (FOR DIF EDGE) |
| PHSR | PHI COMPONENT OF PROPAGATION DIRECTION AFTER REFL IN RCS |
| PP | NEGATIVE DOT PRODUCT OF EDGE BINORMAL AND INCIDENT RAY UNIT NORMA |
| PS | PSR*DPR |
| PSD | DIFFRACTED RAY PHI ANGLE IN EDGE-FIXED COORDINATE SYSTEM |
| PSO | PSOR*DPR |
| PSOD | INCIDENT RAY PHI ANGLE IN EDGE-FIXED COORDINATE SYSTEM |
| PSOR | PHI COMPONENT OF INCIDENT RAY DIRECTION IN EDGE FIXED COORDINATE SYSTEM |
| PSR | PHI COMPONENT OF DIF RAY DIRECTION IN EDGE-FIXED COORD SYS |
| QD | DOT PRODUCT OF PLATE NORMAL AND PROPAGATION DIRECTION |
| QI | NEGATIVE OF DOT PRODUCT OF PLATE NORMAL AND INCIDENT RAY PROPAGATION DIRECTION |
| RM | MAGNITUDE OF VECTOR FROM CORNER MC TO SOURCE |
| RX<br>RY<br>RZ | X,Y, AND Z COMPONENTS OF VECTOR FROM CORNER MC T' SOURCE |
| SBO | SINE OF BO, THE ANGLE THE DIFFRACTED RAY MAKES WITH THE EDGE |
| SNP | SINE OF HALF WEDGE ANGLE |
| SP | DISTANCE FROM SOURCE TO DIFFRACTION POINT (FROM SUB. DFPTWD) |
| SPH | SINE OF PSR |
| SPHJ | SINE OF PHJR |
| SPHO | SINE OF PSOR |
| SPP | DISTANCE FROM SOURCE TO MODIFIED DIFFRACTION POINT |
| STHJ | SINE OF THJR |
| STHR | SINE OF THR |
| TERM | COEFFICIENT OF CORNER DIFFRACTED FIELDS |
| THIR | THETA COMPONENT OF INCIDENT RAY DIRECTION IN REF COORD SYS |
| THJR | THETA COMPONENT OF RAY PROP. DIR. BETWEEN DIF. AND REFLECTION IN RCS |
| THPR | ANGLE DIFFRACTED RAY MAKES WITH EDGE |
| THR | ANGLE BETWEEN EDGE UNIT VECTOR AND RAY FROM SOURCE TO CORNER MC |
| TPP | DISTANCE PARAMETER USED IN CALCULATING DIFFRACTION COEFFICIENTS |
| VECT | VECTOR USED TO MOVE DIFFRACTION POINT OFF EDGE FOR SHADOWING TESTS |
| VI | UNIT VECTOR OF RAY INCIDENT ON EDGE FROM SOURCE (FROM SUBROUTINE DFPTWD) |

| | |
|---|---|
| VIP | UNIT VECTOR FROM SOURCE TO MODIFIED DIFFRACTION POINT |
| VMG | DISTANCE ALONG THE EDGE FROM FIRST CORNER OF EDGE ME TO DIFFRACTION POINT |
| VT | X,Y, AND Z COMPONENTS OF UNIT VECTOR ON PLATE MR NORMAL TO PLANE OF INCIDENCE (TANGENT TO PLATE) |
| VXS | 3X3 MATRIX DEFINING THE SOURCE COORDINATE SYSTEM AXES |
| XD | DIFFRACTION POINT (CALCULATED IN SUB. DFPTWD) |
| XDP | MODIFIED DIFFRACTION POINT USED FOR SHADOWING TESTS<br>ALSO, LOCATION OF DIFF POINT IMAGE IN PLATE MR |
| XDPP | DIFFRACTION POINT, CONVERTED TO REFLECTION HIT POINT |
| XS | SOURCE LOCATION IN REF COORD SYS |
| ZP | DOT PRODUCT OF PROPAGATION UNIT VECTOR AND VECTOR FROM DIFFRACTION POINT TO CORNER MC |

## CODE LISTING

```
C-----------------------------------------------------------------------
      SUBROUTINE DPLRPL(EDTH,EDPH,ECTH,ECPH,FNN,ME,MP,MR)
C!!!
C!!!  DETERMINES THE DIFFRACTED/REFLECTED FIELD WITH PHASE
C!!!  REFERRED TO ORIGIN. RAY IS DIFFRACTED FROM EDGE #ME ON
C!!!  PLATE #MP AND REFLECTED FROM PLATE #MR
C!!!
      COMPLEX EF,EG,EIPR,EIPL,EXPH,DS,DH,DPS,DPH,EDPR,EDPL,EDTH,EDPH
      COMPLEX EIPRP,EIPLP,EIX,EIY,EIZ,CORN,FFCT,A1,A2
      COMPLEX ECTH,ECPH,ECBI,ECBR
      DIMENSION VI(3),XD(3),PHO(3),PH(3),BOP(3),BO(3),XDP(3)
      DIMENSION DJ(3),VT(3),VIP(3),XDPP(3)
      LOGICAL LSURF,LHIT,LDEBUG,LTEST,LSLOPE,LCORNR,LDIF
      COMMON/TEST/LDEBUG,LTEST
      COMMON/LOGDIF/LSLOPE,LCORNR
      COMMON/GEOPLA/X(14,6,3),V(14,6,3),VP(14,6,3),VN(14,3)
     2,MEP(14),MPX
      COMMON/SORINF/XS(3),VXS(3,3)
      COMMON/DIR/D(3),THSR,PHSR,SPHS,CPHS,STHS,CTHS
      COMMON/BNDFCL/BD(14,6,2)
      COMMON/THPHUV/DT(3),DP(2)
      COMMON/PIS/PI,TPI,DPR,RPD
      COMMON/EDMAG/VMAG(14,6)
      COMMON/SURFAC/LSURF(14)
      FN=FNN
      MC=ME+1
      IF (MC.GT.MEP(MP)) MC=1
C!!!  INITIALIZE FIELDS
      EDTH=(0.,0.)
      EDPH=(0.,0.)
      ECTH=(0.,0.)
      ECPH=(0.,0.)
C!!!  1.  COMPUTE INCIDENT DIRECTION OF FIELD ON PLATE #MR
      CALL REFBP(PHJR,THJR,PHSR,THSR,MR)
      SPHJ=SIN(PHJR)
      CPHJ=COS(PHJR)
      STHJ=SIN(THJR)
      CTHJ=COS(THJR)
      DJ(1)=CPHJ*STHJ
      DJ(2)=SPHJ*STHJ
      DJ(3)=CTHJ
C!!!  2. PERFORM DIFFRACTION POINT GEOMETRY CALCULATIONS
C!!!  TO OBTAIN RAY IN DIRECTION DJ
      DV=0.
      DO 10 N=1,3
 10   DV=DV+DJ(N)*V(MP,ME,N)
      IF(ABS(DV).GT.0.999) GO TO 40
      BDEL=0.
      IF (LCORNR) BDEL=0.3
      BDLOW=BD(MP,ME,1)-BDEL
      BDHI=BD(MP,ME,2)+BDEL
C!!!  DETERMINE IF DIFFRACTION EXISTS
      IF (DV.LT.BDLOW.OR.DV.GT.BDHI) GO TO 40
C!!!  COMPUTE EDGE DIFFRACTION PT.
      CALL DFPTWD(XS,DV,VI,SP,XD,ME,MP)
      ADN=0.
      VMG=0.
      AFN=FNN
      IF(AFN.GT.2.)AFN=4.-AFN
      CNP=COS(AFN*PI/2.)
      SNP=SIN(AFN*PI/2.)
      DO 15 N=1,3
      XDP(N)=XD(N)
      VMG=VMG+(XD(N)-X(MP,ME,N))*V(MP,ME,N)
 15   ADN=ADN+(XS(N)-X(MP,1,N))*VN(MP,N)
      LDIF=.TRUE.
```

```
C!!!  IS DIF POINT ON PLATE EDGE?  IF NOT SET DIF POINT AT
C!!!  APPROPRIATE CORNEM AND SET LDIF FALSE
      IF (VMG.LT.1.E-5) GO TO 101
      IF (VMG.LT.VMAG(MP,ME)-1.E-4) GO TO 102
      DO 103 N=1,3
  103 XDP(N)=X(MP,MC,N)-1.E-4*V(MP,ME,N)
      LDIF=.FALSE.
      GO TO 102
  101 DO 104 N=1,3
  104 XDP(N)=X(MP,ME,N)+1.E-4*V(MP,ME,N)
      LDIF=.FALSE.
  102 DO 16 N=1,3
      VECT=VP(MP,ME,N)*CNP+VN(MP,N)*SNP
      XDP(N)=XDP(N)+1.E-5*VECT
 16   XDPP(N)=XDP(N)
C!!!  3.   CHECK TO SEE IF RAY IS SHADOWED ANYWHERE
C!!!  DETERMINE IF REFLECTION OFF PLATE #MR OCCURS
      CALL PLAINT(XDPP,DJ,DHIT,-MR,LHIT)
      IF(.NOT.LHIT) GO TO 40
C!!!  DETERMINE IF RAY AFTER REFLECTION HITS PLATE
      CALL PLAINT(XDPP,D,DHT,MR,LHIT)
      IF(LHIT) GO TO 40
C!!!  DETERMINE IF RAY AFTER REFLECTION HITS CYLINDER
      CALL CYLINT(XDPP,D,PHSR,DHT,LHIT,.TRUE.)
      IF(LHIT) GO TO 40
C!!!  DETERMINE IF EDGE DIF. RAY HITS PLATE BEFORE REFLECTION
      CALL PLAINT(XDP,DJ,DHT,MR,LHIT)
      IF(LHIT.AND.(DHT.LT.DHIT)) GO TO 40
C!!!  DETERMINE IF EDGE DIF. RAY HITS CYLINDER BEFORE REFLECTION
      CALL CYLINT(XDP,DJ,PHJR,DHT,LHIT,.TRUE.)
      IF(LHIT.AND.(DHT.LT.DHIT)) GO TO 40
      SPP=0.
      DO 111 N=1,3
      VIP(N)=XDP(N)-XS(N)
  111 SPP=SPP+VIP(N)*VIP(N)
      SPP=SQRT(SPP)
      DO 112 N=1,3
  112 VIP(N)=VIP(N)/SPP
C!!!  DETERMINE IF RAY FROM SOURCE HITS A PLATE OR A CYLINDER
C!!!  BEFORE DIF.
      CALL PLAINT(XS,VIP,DHT,MP,LHIT)
      IF(LHIT.AND.(DHT.LT.SPP)) GO TO 40
      THIR=BTAN2(SQRT(VI(1)*VI(1)+VI(2)*VI(2)),VI(3))
      PHIR=BTAN2(VI(2),VI(1))
      CALL CYLINT(XS,VI,PHIR,DHT,LHIT,.FALSE.)
      IF(LHIT.AND.(DHT.LT.SPP)) GO TO 40
C!!!  4.  CALCULATE DIFFRACTION ANGLES AND RELATED GEOMETRY
      OI=0.
      PP=0.
      OD=0.
      PD=0.
      DO 20 N=1,3
      OI=OI-VN(MP,N)*VI(N)
      PP=PP-VP(MP,ME,N)*VI(N)
      OD=OD+VN(MP,N)*DJ(N)
 20   PD=PD+VP(MP,ME,N)*DJ(N)
C!!!  CALCULATE PSO AND PS, THE INCIDENT AND DIFFRACTED PHI ANGLES
C!!!  IN EDGE-FIXED COORDINATE SYSTEM
      PSOR=BTAN2(OI,PP)
      PSO=DPR*PSOR
      IF(PSO.LT.0.) PSO=360.+PSO
      PSR=BTAN2(OD,PD)
      PS=DPR*PSR
      IF(PS.LT.0.) PS=360.+PS
      PSOD=PSO
      PSD=PS
      IF(FN.LE.2.)GO TO 21
      FN=FN-2.
      PSOD=360.-PSO
      PSD=360.-PS
21    FNP=FN*180.+1.E-4
      IF(PSO.GT.FNP.OR.PS.GT.FNP) GO TO 40
      SPHO=SIN(PSOR)
      CPHO=COS(PSOR)
      SPH=SIN(PSR)
      CPH=COS(PSR)
C!!!  COMPUTE DIFFRACTION POLARIZATION UNIT VECTORS(PHO,PH,BOP,BO)
      DO 30 N=1,3
      PHO(N)=-VP(MP,ME,N)*SPHO+VN(MP,N)*CPHO
30    PH(N)=-VP(MP,ME,N)*SPH+VN(MP,N)*CPH
      BOP(1)=PHO(2)*VI(3)-PHO(3)*VI(2)
      BOP(2)=PHO(3)*VI(1)-PHO(1)*VI(3)
      BOP(3)=PHO(1)*VI(2)-PHO(2)*VI(1)
      BO(1)=PH(2)*DJ(3)-PH(3)*DJ(2)
      BO(2)=PH(3)*DJ(1)-PH(1)*DJ(3)
      BO(3)=PH(1)*DJ(2)-PH(2)*DJ(1)
C!!!  COMPUTE SBO=SINE(BO)
      SBO=SQRT((V(MP,ME,3)*DJ(2)-V(MP,ME,2)*DJ(3))**2+(V(MP,ME,1)
     2*DJ(3)-V(MP,ME,3)*DJ(1))**2+(V(MP,ME,2)*DJ(1)-V(MP,ME,1)
     2*DJ(2))**2)
      TPP=SP*SBO*SBO
C!!!  5.  CALCULATE EDGE DIFFRACTED FIELDS
C!!!  COMPUTE SOURCE PATTERN FACTORS
      CALL SOURCE(EF,EG,EIX,EIY,EIZ,THIR,PHIR,VXS)
C!!!  COMPUTE INCIDENT FIELD COMPONENTS PARALLEL AND PERP. TO EDGE
      EIPR=EIX*PHO(1)+EIY*PHO(2)+EIZ*PHO(3)
      EIPL=EIX*BOP(1)+EIY*BOP(2)+EIZ*BOP(3)
C!!!  IF SLOPE DIF IS DESIRED, COMPUTE INCIDENT SLOPE FIELD
C!!!  PATTERN FACTORS
      IF(LSLOPE)CALL SOURCP(EIPRP,EIPLP,VI,PHO,BOP,VXS)
C!!!  CALCULATE LOCATION OF DIF POINT IMAGE IN PLATE MR
      CALL IMAGE(XDP,XD,ADN,MR)
C!!!  CALCULATE PHASE TERM FOR DIF IMAGE POINT
      GAM=XDP(1)*D(1)+XDP(2)*D(2)+XDP(3)*D(3)
      EXPH=CEXP(CMPLX(0.,TPI*(GAM-SP)))/SQRT(SP)
C!!!  COMPUTE EDGE DIFFRACTION COEFFICIENTS
      CALL DN(DS,DH,DPS,DPH,TPP,PSD,PSOD,SBO,FN,LSURF(MP))
      IF (LDEBUG) WRITE (6,*) EIPR,EIPL,EIPRP,EIPLP
      IF (LDEBUG) WRITE (6,*) DS,DH,DPS,DPH
      IF (LDEBUG) WRITE (6,*) TPP,PSD,PSOD,SBO,FN
C!!!  COMPUTE PERPENDICULAR AND PARALLEL COMPONENTS OF
C!!!  EDGE DIFFRACTED FIELD (EDPR,EDPL)
      EDPR=-EIPR*DH
      EDPL=-EIPL*DS
C!!!  IF SLOPE DIF IS DESIRED, ADD SLOPE FIELDS TO EDGE DIF
C!!!  FIELD COMPONENTS
      IF(.NOT.LSLOPE)GO TO 201
      EDPR=EDPR-EIPRP*DPH/CMPLX(0.,TPI*SP*SBO)
      EDPL=EDPL-EIPLP*DPS/CMPLX(0.,TPI*SP*SBO)
C!!!  6.  COMPUTE EDGE DIFFRACTED REFLECTED RAY
C!!!  COMPUTE VARIABLES TO TRANSFORM POLARIZATION FROM INCIDENT
C!!!  RAY FIXED COORD SYS TO REFLECTION PLATE COORD SYS
201   VT(1)=VN(MR,2)*DJ(3)-VN(MR,3)*DJ(2)
      VT(2)=VN(MR,3)*DJ(1)-VN(MR,1)*DJ(3)
      VT(3)=VN(MR,1)*DJ(2)-VN(MR,2)*DJ(1)
      C11A=VN(MR,1)*BO(1)+VN(MR,2)*BO(2)+VN(MR,3)*BO(3)
      C12A=VN(MR,1)*PH(1)+VN(MR,2)*PH(2)+VN(MR,3)*PH(3)
      C21A=VT(1)*BO(1)+VT(2)*BO(2)+VT(3)*BO(3)
      C22A=VT(1)*PH(1)+VT(2)*PH(2)+VT(3)*PH(3)
C!!!  COMPUTE VARIABLES TO TRANSFORM RAY POLARIZATION FROM PLATE
C!!!  COORDINATES TO RAY-FIXED THETA AND PHI COMPONENTS FOR REFL
C!!!  RAY IN MCS
```

```
          C11=VN(MR,1)*DT(1)+VN(MR,2)*DT(2)+VN(MR,3)*DT(3)
          C12=VN(MR,1)*DP(1)+VN(MR,2)*DP(2)
          C21=VT(1)*DT(1)+VT(2)*DT(2)+VT(3)*DT(3)
          C22=VT(1)*DP(1)+VT(2)*DP(2)
          A3=C11*C22-C12*C21
    C!!!  DETERMINE IF EDGE DIF FIELD EXISTS
          IF (.NOT.LDIF) GO TO 202
    C!!!  COMPUTE POLARIZATION OF DIF FIELD INCIDENT ON PLATE MR
    C!!!  IN COMPONENTS NORMAL AND TANGENT TO THE PLATE (A1 AND A2)
          A1=EDPL*C11A+EDPR*C12A
          A2=EDPL*C21A+EDPR*C22A
    C!!!  CALCULATE THETA AND PHI COMPONENTS OF REFL FIELD IN RCS
          EF=(A1*C22+A2*C12)/A3
          EG=-(A2*C11+A1*C21)/A3
    C!!!  ADD PHASE TERM TO OBTAIN DIF REFL FIELD COMPONENTS
    C!!!  EDTH AND EDPH
          EDTH=EF*EXPH
          EDPH=EG*EXPH
    C!!!  7. IF CORNER DIF FIELD IS DESIRED, COMPUTE CORNER FIELDS
     202  IF (.NOT.LCORNR) GO TO 40
          BETN=PSD-PSOD
          BETP=PSD+PSOD
          MC=ME-1
          ISN=1
    C!!!  LOOP THRU BOTH CORNERS ON EDGE #ME
    35    MC=MC+1
          IF(MC.GT.NE(MP)) MC=1
          ISN=-ISN
          RX=XS(1)-X(MP,MC,1)
          RY=XS(2)-X(MP,MC,2)
          RZ=XS(3)-X(MP,MC,3)
          RM=SQRT(RX*RX+RY*RY+RZ*RZ)
          CTH=V(MP,ME,1)*RX+V(MP,ME,2)*RY+V(MP,ME,3)*RZ
          CTH=ISN*CTH/RM
          CTHP=ISN*DV
          THPR=ACOS(CTHP)
          THR=ACOS(CTH)
          STHR=SIN(THR)
          DEL=2.*TPI*RM*(COS(.5*(THR+THPR))**2)
          ZP=(X(MP,MC,1)-XD(1))*DJ(1)+(X(MP,MC,2)-XD(2))*DJ(2)
         2+(X(MP,MC,3)-XD(3))*DJ(3)
          TERM=-STHR/TPI/(CTH+CTHP)/SQRT(RM)
    C!!!  COMPUTE CORNER DIFFRACTION COEFFICIENT (CORN).
          CORN=-TERM*FFCT(DEL)*CEXP(CMPLX(0.,-TPI*(RM-SP-ZP)-.25*PI))
          CALL DI(ECBI,TPP,BETN,SBO,FN,DEL,.TRUE.)
          IF(LSURF(MP))GO TO 311
          CALL DI(ECBR,TPP,BETP,SBO,FN,DEL,.TRUE.)
    C!!!  COMPUTE MODIFIED EDGE DIFF. COEFFICIENTS (DH,DS).
          DH=ECBI+ECBR
          DS=ECBI-ECBR
          GO TO 312
    311   DH=ECBI
          DS=(0.,0.)
    C!!!  COMPUTE COMPONENTS OF CORNER DIFFRACTED FIELD PARALLEL
    C!!!  AND PERPENDICULAR TO EDGE
    312   EDPR=-EIPR*DH*EXPH
          EDPL=-EIPL*DS*EXPH
          IF(.NOT.LSLOPE)GO TO 203
          EDPR=EDPR-EIPRP*DPH*EXPH/CMPLX(0.,TPI*SP*SBO)
          EDPL=EDPL-EIPLP*DPS*EXPH/CMPLX(0.,TPI*SP*SBO)
    C!!!  COMPUTE CORNER DIFFRACTED FIELDS INCIDENT ON PLATE MR IN
    C!!!  PLATE COORDINATE SYSTEM
    203   A1=EDPL*C11A+EDPR*C12A
          A2=EDPL*C21A+EDPR*C22A
    C!!!  COMPUTE CORNER DIFFRACTED FIELDS AFTER REFLECTION IN RCS
          EF=(A1*C22+A2*C12)/A3
```

```
      EG=-(A2*C11+A1*C21)/A3
C!!!  COMPUTE THETA AND PHI COMPONENTS OF CORNER DIFFRACTED
C!!!  REFLECTED FIELDS (ECTH, ECPH) IN RCS
      ECTH=ECTH+EF*CORN
      ECPH=ECPH+EG*CORN
      IF (.NOT.LDEBUG) GO TO 36
      WRITE (6,*) DS,DH,EDPR,EDPL
      WRITE (6,*) ECTH,ECPH,CORN
      WRITE (6,*) EF,EG
 36   CONTINUE
      IF(MC.EQ.ME) GO TO 35
  40  IF (.NOT.LTEST) GO TO 204
      WRITE (6,205)
  205 FORMAT (/,' TESTING DPLRPL SUBROUTINE')
      WRITE (6,*) EDTH,EDPH,ECTH,ECPH
      WRITE (6,*) FN,ME,MP,MR
  204 RETURN
      END
```

## DPTNFW

PURPOSE

To compute the diffraction point for a ray which is diffracted by a given edge and observed at a specified near field point of the plate.

PERTINENT GEOMETRY

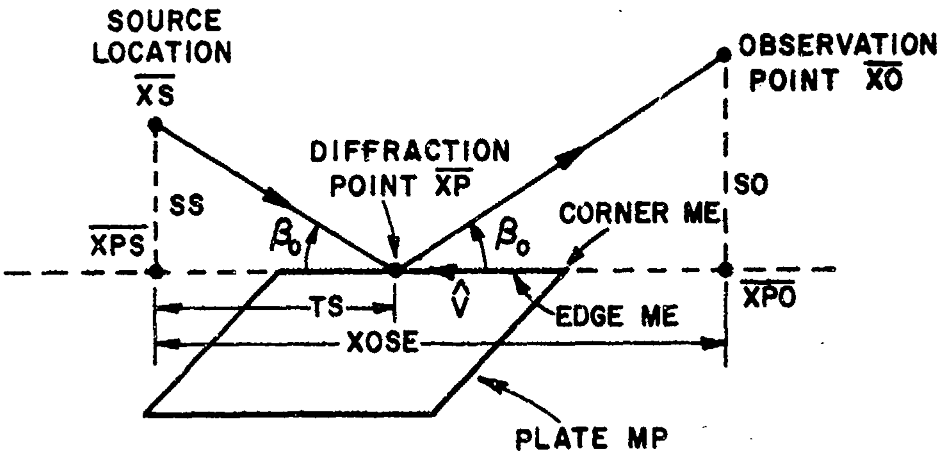

SOURCE LOCATION $\overline{XS}$
OBSERVATION POINT $\overline{XO}$
DIFFRACTION POINT $\overline{XP}$
SS
SO
$\overline{XPS}$
$\beta_o$
$\beta_o$
CORNER ME
$\hat{V}$
$\overline{XPO}$
TS
EDGE ME
XOSE
PLATE MP

Figure 60-- Geometry for finding the diffraction point with the observation point in the near field of the plate.

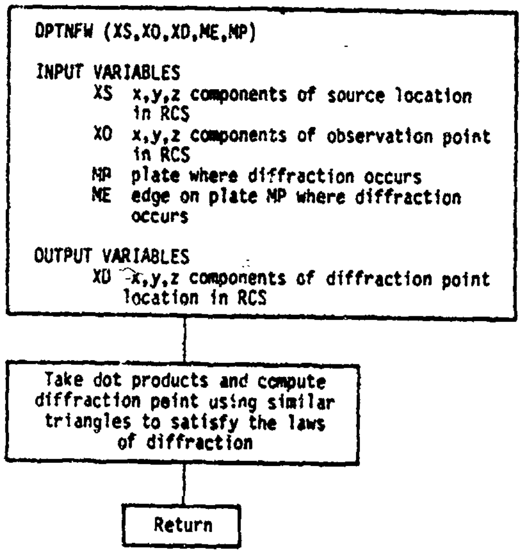
METHOD

The diffraction point is found using similar triangles defined by perpendiculars from the source and observation points to the edge line. The diffraction point is given by

$$\overline{XD} = \overline{XPS} + \frac{SSxXOSE}{SS + SO} \hat{V},$$

where the above quantities are illustrated in Figure 60.

## FLOW DIAGRAM



OPTNFW (XS,XO,XD,ME,MP)

INPUT VARIABLES
XS x,y,z components of source location in RCS
XO x,y,z components of observation point in RCS
MP plate where diffraction occurs
ME edge on plate MP where diffraction occurs

OUTPUT VARIABLES
XD x,y,z components of diffraction point location in RCS

Take dot products and compute diffraction point using similar triangles to satisfy the laws of diffraction

Return

## SYMBOL DICTIONARY

| | |
|---|---|
| SS | DISTANCE FROM SOURCE TO POINT XPS |
| SO | DISTANCE FROM OBSERVATION POINT TO POINT XPO |
| TS | DISTANCE FROM XPS TO XD ALONG EDGE LINE |
| XOCE | DOT PRODUCT OF RAY FROM CORNER ME TO OBSERVATION POINT AND EDGE UNIT VECTOR |
| XOSE | DOT PRODUCT OF RAY FROM SOURCE TO OBSERVATION POINT AND EDGE UNIT VECTOR |
| XPS | POINT ON LINE THROUGH EDGE ME CLOSEST TO SOURCE |
| XPO | POINT ON LINE THROUGH EDGE ME CLOSEST TO OBSERVATION POINT |
| XSCE | DOT PRODUCT OF RAY FROM CORNER ME TO SOURCE AND EDGE UNIT VECTOR |

## CODE LISTING

```
C-----------------------------------------------------------------------
      SUBROUTINE DPTNFW(XS,XO,XD,ME,MP)
C!!!
C!!!  DETERMINES THE NEAR FIELD DIFFRACTION POINT ON A PLATE EDGE
C!!!
      DIMENSION XS(3),XO(3),XPS(3),XPO(3),XD(3)
      COMMON/GEOPLA/X(14,6,3),V(14,6,3),VP(14,6,3),VN(14,3)
     2,MEP(14),MPX
      XSCE=Ø.
      XOCE=Ø.
      XOSE=Ø.
      DO 1Ø N=1,3
      XSCE=XSCE+(XS(N)-X(MP,ME,N))*V(MP,ME,N)
      XOCE=XOCE+(XO(N)-X(MP,ME,N))*V(MP,ME,N)
   1Ø XOSE=XOSE+(XO(N)-XS(N))*V(MP,ME,N)
      DO 2Ø N=1,3
      XPS(N)=XSCE*V(MP,ME,N)+X(MP,ME,N)
   2Ø XPO(N)=XOCE*V(MP,ME,N)+X(MP,ME,N)
      SS=(XS(1)-XPS(1))*(XS(1)-XPS(1))+(XS(2)-XPS(2))*(XS(2)-XPS(2))
     2+(XS(3)-XPS(3))*(XS(3)-XPS(3))
      SS=SQRT(SS)
      SO=(XO(1)-XPO(1))*(XO(1)-XPO(1))+(XO(2)-XPO(2))*(XO(2)-XPO(2))
     2+(XO(3)-XPO(3))*(XO(3)-XPO(3))
      SO=SQRT(SO)
      TS=SS*XOSE/(SS+SO)
      DO 3Ø N=1,3
   3Ø XD(N)=XPS(N)+TS*V(MP,ME,N)
      RETURN
      END
```

## DQG32

### PURPOSE

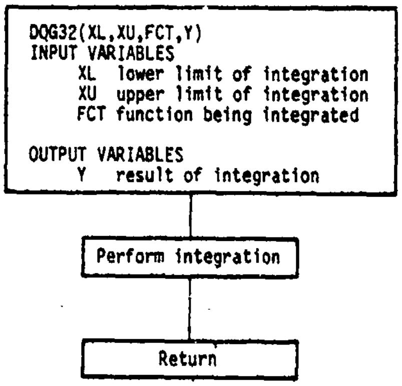
To numerically integrate a given function over a specified range.

### METHOD

This subroutine uses a 32 point Gaussian quadrature formula to compute the integral of a function[11]. The form of the integral is given as

$$Y = \int_{XL}^{XU} FCT(x)dx \ .$$

### FLOW DIAGRAM



### SYMBOL DICTIONARY

| | |
|---|---|
| FCT | FUNCTION DEFINING THE INTEGRAND |
| XL | LOWER BOUND OF INTEGRAL |
| XU | UPPER BOUND OF INTEGRAL |
| Y | RESULT OF INTEGRAL |

## CODE LISTING

```
C-----------------------------------------------------------------------
      SUBROUTINE DCG32(XL,XU,FCT,Y)
C!!!
C!!!  32 POINT GAUSSIAN QUADRATURE INTEGRATION ROUTINE
C!!!
      A=.5DØ*(XU+XL)
      B=XU-XL
      C=.49863193Ø92474DØ*B
      Y=.35Ø93Ø5ØØ47350D-2*(FCT(A+C)+FCT(A-C))
      C=.49280575577263DØ*B
      Y=Y+.8137197365452D-2*(FCT(A+C)+FCT(A-C))
      C=.48238112779375DØ*B
      Y=Y+.12696Ø32654631D-1*(FCT(A+C)+FCT(A-C))
      C=.467453Ø3796886DØ*B
      Y=Y+.1713693145651ØD-1*(FCT(A+C)+FCT(A-C))
      C=.44816057788302DØ*B
      Y=Y+.214179490111113D-1*(FCT(A+C)+FCT(A-C))
      C=.4246838Ø686628DØ*B
      Y=Y+.25499029631188D-1*(FCT(A+C)+FCT(A-C))
      C=.39724189798397DØ*B
      Y=Y+.29342Ø46739267D-1*(FCT(A+C)+FCT(A-C))
      C=.366091Ø5937Ø14DØ*B
      Y=Y+.32911111388180D-1*(FCT(A+C)+FCT(A-C))
      C=.3315221334651ØDØ*B
      Y=Y+.36172897Ø54424D-1*(FCT(A+C)+FCT(A-C))
      C=.29385787862Ø38DØ*B
      Y=Y+.39Ø969478935350D-1*(FCT(A+C)+FCT(A-C))
      C=.25344995446611DØ*B
      Y=Y+.41655962113473D-1*(FCT(A+C)+FCT(A-C))
      C=.21Ø6756380653IDØ*B
      Y=Y+.43826Ø465Ø22Ø1D-1*(FCT(A+C)+FCT(A-C))
      C=.1659343Ø114106DØ*B
      Y=Y+.45586939347881D-1*(FCT(A+C)+FCT(A-C))
      C=..11964368112606DØ*B
      Y=Y+.46922199540402D-1*(FCT(A+C)+FCT(A-C))
      C=.7223598079139D-1*B
      Y=Y+.47819360039637D-1*(FCT(A+C)+FCT(A-C))
      C=.24153832843869D-1*B
      Y=B*(Y+.48270044257363D-1*(FCT(A+C)+FCT(A-C)))
      RETURN
      END
```

## DW

### PURPOSE

To determine wedge and slope diffraction coefficients for the soft and hard boundary conditions.

### METHOD

This subroutine calculates the edge diffraction and slope diffraction coefficients for the hard and soft boundary conditions using the Uniform Geometrical Theory of Diffraction[4,5]. The edge diffraction coefficient has the form

$$D_{\substack{h\\s}} = DI(R,\phi-\phi',\sin\beta_o,n) \pm DI(R,\phi+\phi',\sin\beta_o,n),$$

where $D_h$ is for the hard case and $D_s$ is for the soft case and n is the wedge angle number (FN).

The slope diffraction coefficient has the form

$$\frac{\partial D_{\substack{h\\s}}}{\partial\phi'} = DPI(R,\phi-\phi',\sin\beta_o,n) \mp DPI(R,\phi+\phi',\sin\beta_o,n).$$

In both cases the $\phi-\phi'$ part refers to the incident part of the diffraction coefficient and $\phi+\phi'$ refers to the reflection part. For grazing incidence where $\phi'=0$, the diffraction coefficients have the form

$$D_h = DI(L,\phi,\sin\beta_o,n)$$

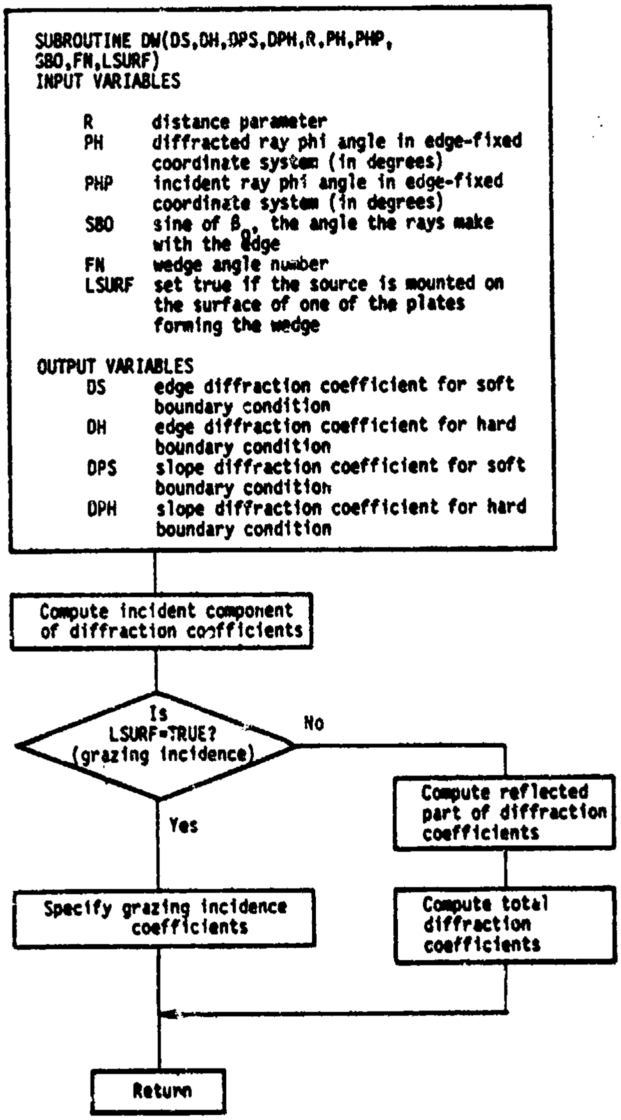
$$D_s = 0$$

$$\frac{\partial D_s}{\partial\phi'} = DPI(L,\phi,\sin\beta_o,n)$$

$$\frac{\partial D_h}{\partial\phi'} = 0 .$$

An illustration of the wedge geometry is given in Figure 55.

## FLOW DIAGRAM



SUBROUTINE DW(DS,DH,DPS,DPH,R,PH,PHP,
SBO,FN,LSURF)
INPUT VARIABLES

- R distance parameter
- PH diffracted ray phi angle in edge-fixed coordinate system (in degrees)
- PHP incident ray phi angle in edge-fixed coordinate system (in degrees)
- SBO sine of $\beta_o$, the angle the rays make with the edge
- FN wedge angle number
- LSURF set true if the source is mounted on the surface of one of the plates forming the wedge

OUTPUT VARIABLES

- DS edge diffraction coefficient for soft boundary condition
- DH edge diffraction coefficient for hard boundary condition
- DPS slope diffraction coefficient for soft boundary condition
- DPH slope diffraction coefficient for hard boundary condition

Compute incident component of diffraction coefficients
Is LSURF=TRUE? (grazing incidence)
No
Yes
Compute reflected part of diffraction coefficients
Compute total diffraction coefficients
Specify grazing incidence coefficients
Return

## SYMBOL DICTIONARY

| | |
|---|---|
| BETN | DIFFERENCE BETWEEN DIFFRACTION AND INCIDENCE ANGLE |
| BETP | DIFFERENCE BETWEEN DIFFRACTION AND IMAGE OF INCIDENCE ANGLE |
| DH | EDGE DIFFRACTION COEFFICIENT FOR THE HARD BOUNDARY CASE |
| DIN | INCIDENT PART OF EDGE DIFFRACTION COEFFICIENT |
| DIP | REFLECTION PART OF EDGE DIFFRACTION COEFFICIENT |
| DPH | SLOPE DIFFRACTION COEFFICIENT FOR THE HARD BOUNDARY CASE |
| DPN | INCIDENT PART OF SLOPE DIFFRACTION COEFFICIENT |
| DPP | REFLECTION PART OF SLOPE DIFFRACTION COEFFICIENT |
| DPS | SLOPE DIFFRACTION COEFFIECIENT FOR THE SOFT BOUNDARY CASE |
| DS | EDGE DIFFRACTION COEFFICIENT FOR THE SOFT BOUNDARY CASE |
| FN | WEDGE ANGLE NUMBER |
| LSURF | A LOGICAL VARIABLE THAT IS SET TRUE IF THE SOURCE IS MOUNTED ON THE SURFACE OF THE WEDGE (GRAZING INCIDENCE) |
| PH | DIFFRACTED RAY PHI ANGLE IN DEGREES |
| PHP | INCIDENT RAY PHI ANGLE IN DEGREES |
| R | DISTANCE PARAMETER |
| SBO | SIN(BO) |

## CODE LISTING

```
C-----------------------------------------------------------------
      SUBROUTINE DW(DS,DH,DPS,DPH,R,PH,PHP,SBO,FN,LSURF)
C!!!
C!!!  WEDGE DIFFRACTION AND SLOPE DIFFRACTION COEFFICIENT
C!!!  FOR THE SOFT AND HARD BOUNDARY CONDITIONS
C!!!
      LOGICAL LSURF
      COMPLEX DIN,DIP,DPN,DPP,DS,DH,DPS,DPH
C!!!  INCIDENT PART OF DIFFRACTION COEFFICIENT
      BETN=PH-PHP
      CALL DI(DIN,R,BETN,SBO,FN,1.,.FALSE.)
      CALL DPI(DPN,R,BETN,SBO,FN)
      IF(.NOT.LSURF)GO TO 10
C!!!  GRAZING INCIDENCE CASE
      DS=(0.,0.)
      DH=DIN
      DPS=DPN
      DPH=(0.,0.)
      RETURN
   10 CONTINUE
C!!!  REFLECTION PART OF DIFFRACTION COEFFICIENT
      BETP=PH+PHP
      CALL DI(DIP,R,BETP,SBO,FN,1.,.FALSE.)
      CALL DPI(DPP,R,BETP,SBO,FN)
      DS=DIN-DIP
      DH=DIN+DIP
      DPS=DPN+DPP
      DPH=DPN-DPP
      RETURN
      END
```

## DZ

### PURPOSE

To compute the diffraction coefficient for an edge formed by two curved surfaces.

### METHOD

This subroutine computes the diffraction coefficient for a curved edge based on the uniform Geometrical Theory of Diffraction [4 ]. The diffraction coefficient is given by

$$D_{\substack{s\\h}}(\phi,\phi',\beta_0) = \frac{e^{-j\pi/4}}{2n\sqrt{2\pi k}\,\sin\beta_0}\left[\frac{2\sin(\pi/n)F[kL^i a(\phi-\phi')]}{\cos(\pi/n)-\cos[(\phi-\phi')/n]}\right.$$

$$\pm\left\{\cot\left(\frac{\pi+(\phi+\phi')}{2n}\right) F[kL^{rn}a^+(\phi+\phi')]\right.$$

$$\left.+\cot\left(\frac{\pi-(\phi+\phi')}{2n}\right)F[kL^{ro}a(\phi+\phi')]\right\} ,$$

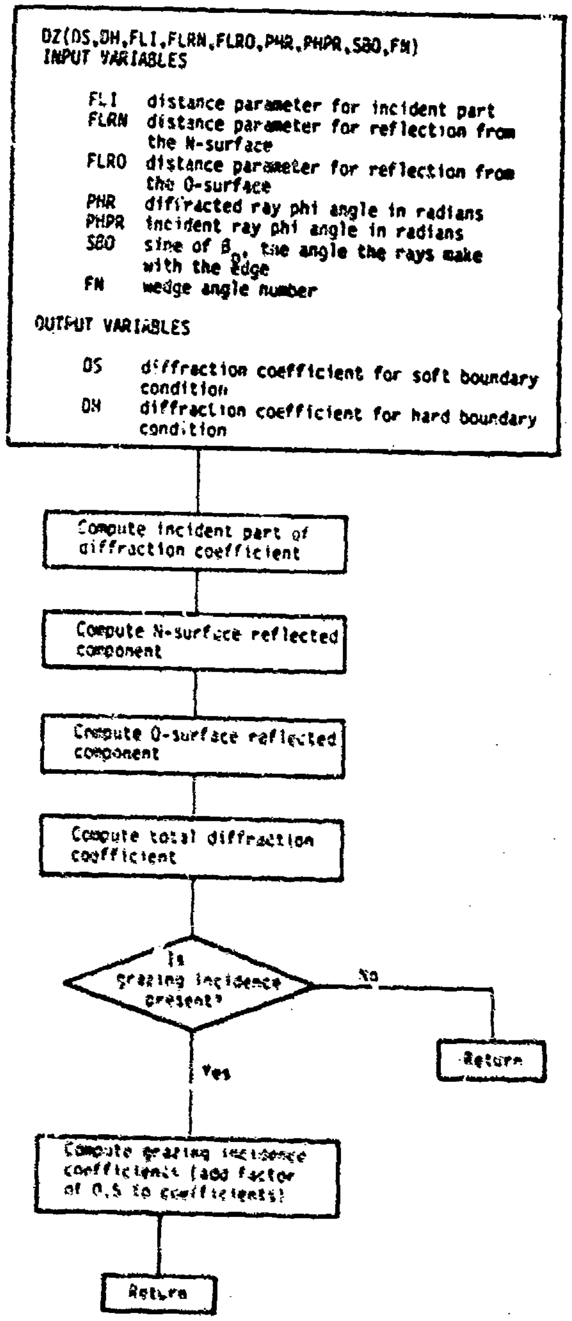
where $a(\beta) = 2\cos^2\beta/2$, $a^+(\beta)=2\cos^2(2\pi n-\beta)/2$, n is the wedge number (FN), and $L^i, L^{rn}, L^{ro}$ are the distance parameters for the incident part, reflection from the n-surface and o-surface, respectively.

When the diffraction angle is close to one of the shadow boundaries, the following approximation is used

$$\cot\left(\frac{\pi\pm\beta}{2n}\right) F[kLa^{\pm}(\beta)] = \pm n\sqrt{2\pi kL}\, e^{j\pi/4}\, e^{jk|L|a^{\pm}} ,$$

where the plus or minus sign is chosen depending on which side of the shadow boundary the diffraction angle is on.

## FLOW DIAGRAM


DZ(DS,DH,FLI,FLRN,FLRO,PHR,PHPR,SBO,FN)
INPUT VARIABLES
FLI distance parameter for incident part
FLRN distance parameter for reflection from the N-surface
FLRO distance parameter for reflection from the O-surface
PHR diffracted ray phi angle in radians
PHPR incident ray phi angle in radians
SBO sine of $\beta_o$, the angle the rays make with the edge
FN wedge angle number
OUTPUT VARIABLES
DS diffraction coefficient for soft boundary condition
DH diffraction coefficient for hard boundary condition
Compute incident part of diffraction coefficient
Compute N-surface reflected component
Compute O-surface reflected component
Compute total diffraction coefficient
Is grazing incidence present?
No
Return
Yes
Compute grazing incidence coefficients (add factor of 0.5 to coefficients)
Return

## SYMBOL DICTIONARY

| | |
|---|---|
| A | ANGLE FUNCTION FOR INCIDENT AND O-SURFACE TRANSITION FUNCTIONS |
| AP | ANGLE FUNCTION FOR N-SURFACE TRANSITION FUNCTION |
| CSP | COS(PMR/2.) |
| DH | DIFFRACTION COEFFICIENT FOR HARD BOUNDARY CONDITION |
| DS | DIFFRACTION COEFFICIENT FOR SOFT BOUNDARY CONDITION |
| F1 | CONSTANT FACTOR |
| F2 | INCIDENT PART OF DIFFRACTION COEFFICIENT |
| F3 | N-SURFACE PART OF DIFFRACTION COEFFICIENT |
| F4 | O-SURFACE PART OF DIFFRACTION COEFFICIENT |
| FLI | DISTANCE PARAMETER FOR THE INCIDENT COMPONENT |
| FLRN | DISTANCE PARAMETER FOR THE REFLECTION FROM THE N-SURFACE |
| FLRO | DISTANCE PARAMETER FOR THE REFLECTION FROM THE O-SURFACE |
| FN | WEDGE ANGLE NUMBER |
| PHPR | INCIDENT RAY ANGLE IN RADIANS |
| PHR | DIFFRACTED RAY ANGLE IN RADIANS |
| PMR | DIFFERENCE BETWEEN DIFFRACTION ANGLE AND THE INCIDENCE ANGLE |
| PPR | DIFFERENCE BETWEEN DIFFRACTION ANGLE AND THE IMAGE OF THE INCIDENCE ANGLE |
| SBO | SINE OF BO |
| TAN1 | N-SURFACE ANGULAR DEPENDENCE OF DIFFRACTION COEFFICIENT |
| TAN2 | O-SURFACE ANGULAR DEPENDENCE OF DIFFRACTION COEFFICIENT |

## CODE LISTING

```
C-----------------------------------------------------------------------
      SUBROUTINE DZ(DS,DH,FLI,FLRN,FLRO,PHR,PHPR,SBO,FN)
C!!!
C!!!  CURVED EDGE DIFFRACTION COEFFICIENT
C!!!
      COMPLEX FKY,F1,F2,F3,F4,DS,DH,CJ
      COMMON/PIS/PI,TPI,DPR,RPD
      PPR=PHR+PHPR
      PMR=PHR-PHPR
      F1=CEXP(CMPLX(0.,-PI/4.))/(2.*FN*TPI*SBO)
C!!!  INCIDENT PART
      CSP=COS(.5*PMR)
      A=2.*CSP*CSP
      IF(ABS(PMR-PI).LT.1.E-5) GO TO 10
      F2=CMPLX(COS(PI/FN)-COS(PMR/FN),0.)
      F2=2.*SIN(PI/FN)*FKY(FLI,A)/F2
      GO TO 15
 10   F2=CEXP(CMPLX(0.,PI/4.+TPI*ABS(FLI)*A))
      IF(CSP.LT.0.) F2=-F2
      F2=-F2*FN*TPI*CSQRT(CMPLX(FLI,0.))
C!!!  N-SURFACE REFLECTION PART
 15   CSP=COS(.5*(TPI*FN-PPR))
      AP=2.*CSP*CSP
      TAN1=TAN((PI+PPR)/(2.*FN))
      IF(ABS(TAN1).LT.1.E-5) GO TO 20
      F3=FKY(FLRN,AP)/TAN1
      GO TO 25
 20   F3=CEXP(CMPLX(0.,PI/4.+TPI*ABS(FLRN)*AP))
      IF(CSP.LT.0.) F3=-F3
      F3=-F3*FN*TPI*CSQRT(CMPLX(FLRN,0.))
C!!!  O-SURFACE REFLECTION PART
 25   CSP=COS(.5*PPR)
      A=2.*CSP*CSP
      TAN2=TAN((PI-PPR)/(2.*FN))
      IF(ABS(TAN2).LT.1.E-5) GO TO 30
      F4=FKY(FLRO,A)/TAN2
      GO TO 35
 30   F4=CEXP(CMPLX(0.,PI/4.+TPI*ABS(FLRO)*A))
      IF(CSP.LT.0.) F4=-F4
      F4=-F4*FN*TPI*CSQRT(CMPLX(FLRO,0.))
C!!!  TOTAL DIFFRACTION COEFFICIENT
 35   DS=F1*(F2+F3+F4)
      DH=F1*(F2-F3-F4)
C!!!  GRAZING INCIDENCE CASE
      IF(PHPR.GT.1.E-5) GO TO 40
      DS=.5*DS
      DH=.5*DH
 40   CONTINUE
      RETURN
      END
```

ENDIF

PURPOSE

To compute the fields due to the diffraction of source fields from a given cylinder end cap edge.

PERTINENT GEOMETRY

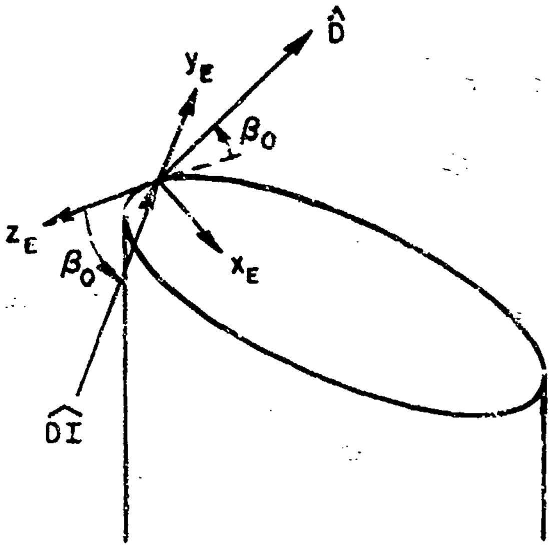


Figure 61-- Illustration of diffraction point coordinate system.

$$\hat{x}_E = \hat{x}\ XEX + \hat{y}\ XEY + \hat{z}\ XEZ$$

$$\hat{y}_E = \hat{x}\ YEX + \hat{y}\ YEY + \hat{z}\ YEZ$$

$$\hat{z}_E = \hat{x}\ ZEX + \hat{y}\ ZEY + \hat{z}\ ZEZ$$

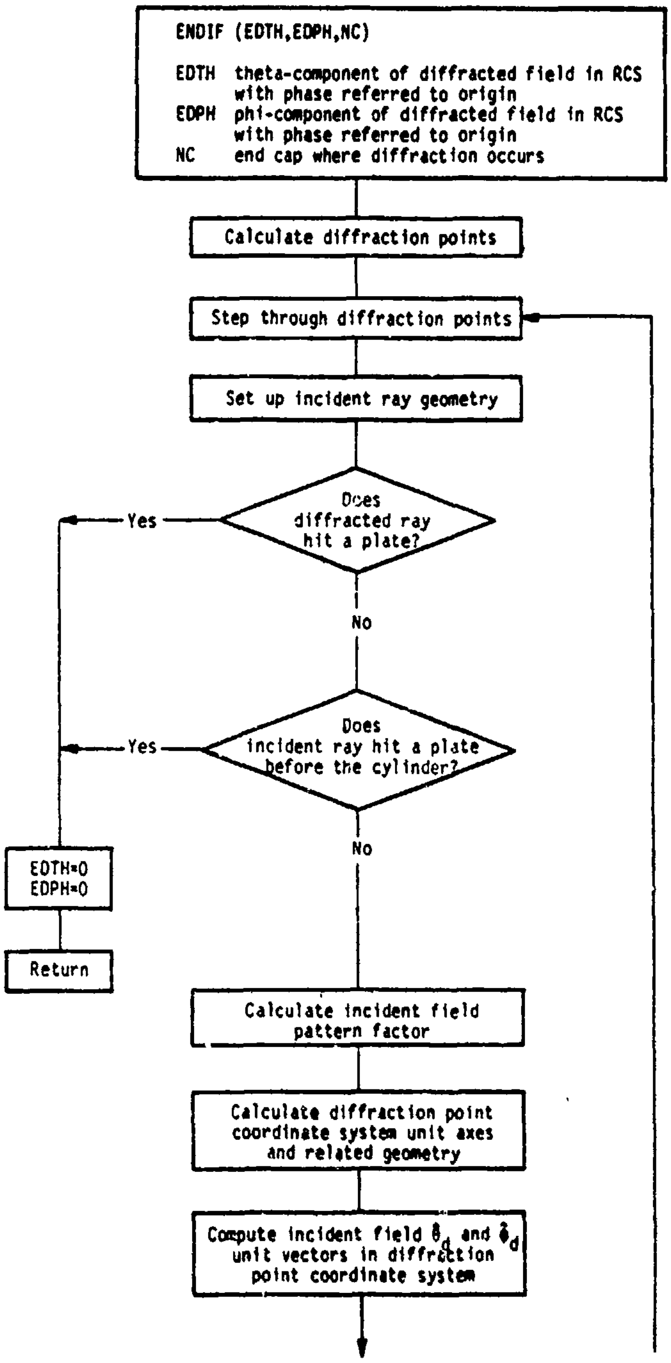
METHOD

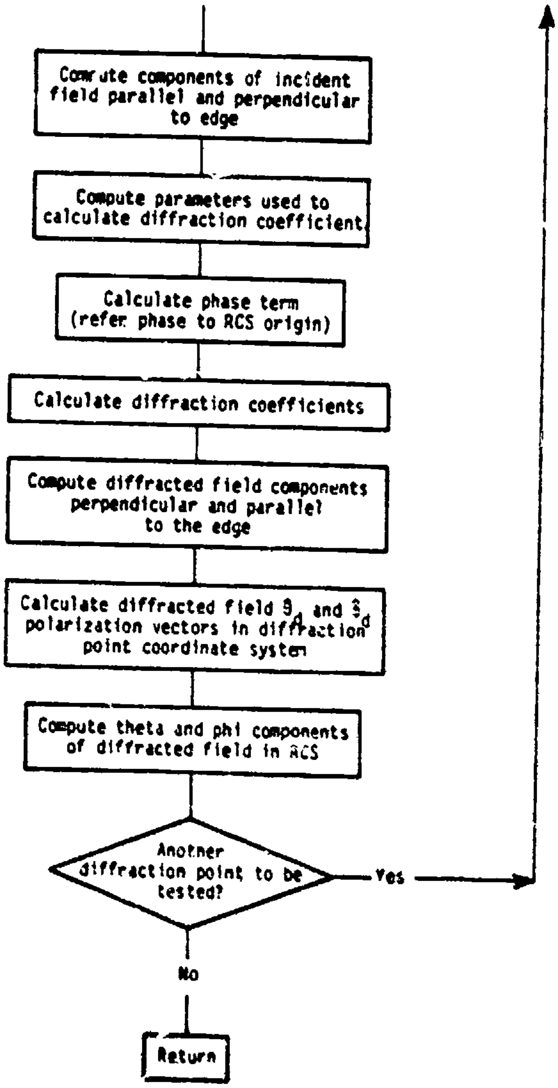
The Geometrical Theory of Diffraction [4]is used to compute the fields diffracted by the curved edges formed by the end cap disk and the curved surface of the elliptic cylinder. The form of diffraction coefficients for the curved edge are similar to that given in subroutine DIFPLT except that the distance parameters and spread factors are slightly different. The details are given on pages 127-131 of Reference 1. The fields from four possible diffraction points on the edge are superimposed to give the total diffracted field from one end cap. For small regions of the radiation pattern, it is posisble that three of the diffraction points will coalesce into one point leaving two diffraciton points on the edge. When this happens a finite spike (psuedo caustic) of small angular extent appears in the pattern. One way to correct for this is by the use of an equivalent current solution[12]. However, this is costly in terms of computation time so it has not been included at present. The overall solution is not effected significantly by this approximation. The phases of the diffracted fields are referred to the reference coordinate system origin and the total field are represented as

$$\bar{E}^{d}_{endcap} = W_m(EDTH\hat{\theta} + EDPH\hat{\phi})\,\frac{e^{-jkR}}{R} ,$$

where the factor $\frac{e^{-jkR}}{R}$ and the source weight ($W_m$) are added elsewhere in the code.

## FLOW DIAGRAM


ENDIF (EDTH,EDPH,NC)
EDTH theta-component of diffracted field in RCS with phase referred to origin
EDPH phi-component of diffracted field in RCS with phase referred to origin
NC end cap where diffraction occurs
Calculate diffraction points
Step through diffraction points
Set up incident ray geometry
Does diffracted ray hit a plate?
Yes
No
Does incident ray hit a plate before the cylinder?
Yes
No
EDTH=0
EDPH=0
Return
Calculate incident field pattern factor
Calculate diffraction point coordinate system unit axes and related geometry
Compute incident field θ̂d and φ̂d unit vectors in diffraction point coordinate system


Comrute components of incident field parallel and perpendicular to edge
Compute parameters used to calculate diffraction coefficient
Calculate phase term (refer phase to RCS origin)
Calculate diffraction coefficients
Compute diffracted field components perpendicular and parallel to the edge
Calculate diffracted field $\hat{\beta}_d$ and $\hat{\phi}_d$ polarization vectors in diffraction point coordinate system
Compute theta and phi components of diffracted field in RCS
Another diffraction point to be tested?
Yes
No
Return

## SYMBOL DICTIONARY

| | |
|---|---|
| AE | RADIUS OF CURVATURE OF EDGE AT DIFFRACTION POINT IN END CAP PLANE |
| BO | THE ANGLE THE INCIDENT (AND DIFFRACTED) RAY MAKES WITH THE EDGE UNIT VECTOR |
| CBO | COSINE OF BO (DOT PRODUCT OF DIFF RAY AND Z AXIS OF DIFFRACTION POINT COORD SYS) |
| CPE | COSINE OF PHER |
| CTE | COSINE OF THER |
| CTHI | DOT PRODUCT OF INCIDENT RAY PROPAGATION DIRECTION UNIT VECTOR AND CYLINDER UNIT NORMAL |
| CV | COSINE OF VR |
| D | X,Y,Z COMPONENTS OF PROPAGATION DIRECTION AFTER DIFFRACTION IN RCS |
| DH | DIFFRACTION COEF FOR HARD BOUNDARY CONDITION |
| DHIT | DISTANCE FROM SOURCE TO NEAREST HIT (FROM PLAINT) |
| DI | X,Y,Z COMPONENTS OF UNIT VECTOR OF INCIDENT RAY PROPAGATION DIRECTION IN RCS |
| DS | DIFFRACTION COEF. FOR SOFT BOUNDARY CONDITION |
| EDPH | PHI COMPONENT OF DIFFRACTED E FIELD IN RCS WITH PHASE REFERRED TO RCS ORIGIN |
| EDPP | COMPONENT OF DIFFRACTED FIELD PARALLEL TO EDGE |
| EDPR | COMPONENT OF DIFFRACTED FIELD PERPENDICULAR TO EDGE |
| EDTH | THETA COMPONENT OF DIFFRACTED E FIELD IN RCS WITH PHASE REFERRED TO RCS ORIGIN |
| EF | THETA COMPONENT OF INCIDENT FIELD PATTERN FACTOR IN RCS |
| EG | PHI COMPONENT OF INCIDENT FIELD PATTERN FACTOR IN RCS |
| EIPP | COMPONENT OF INCIDENT E FIELD PARALLEL TO EDGE |
| EIPR | COMPONENT OF INCIDENT E FIELD PERPENDICULAR TO EDGE |
| EIX<br>EIY<br>EIZ | X,Y,Z COMPONENTS OF INCIDENT FIELD PATTERN FACTOR |
| EN | NORMALIZATION CONSTANT FOR Z AXIS OF DIF POINT COORD SYS |
| EX<br>EY<br>EZ | X,Y,Z COMPONENTS DEFINING UNIT EDGE VECTOR (Z AXIS OF DIFFRACTION POINT COORD SYS) |
| FN | WEDGE ANGLE NUMBER |
| I | DO LOOP VARIABLE |
| LHIT | SET TRUE IF RAY HITS A PLATE (FROM PLAINT) |
| NC | END CAP WHERE DIFFRACTION OCCURS |
| NCC | SIGN CHANGE VARIABLE |
| PH | COMPLEX PHASE COEFFICIENT |
| PHEDR | PHI COMPONENT OF DIFFRACTED RAY DIRECTION IN DIFFRACTION POINT COORDINATE SYSTEM |
| PHER | PHI COMPONENT OF INCIDENT RAY PROPAGATION DIRECTION IN DIFFRACTION POINT COORDINATE SYSTEM |
| PHEX<br>PHEY<br>PHEZ | POLARIZATION UNIT VECTOR IN PHI DIRECTION FOR INC. OR DIFFRACTED RAY IN DIFFRACTION POINT COORDINATE SYSTEM IN (X,Y,Z) RCS COMPONENTS |
| PHIR | PHI COMPONENT OF INCIDENT RAY DIRECTION IN RCS |
| RG | RADIUS OF CURVATURE OF CYLINDER SURFACE AT DIFF POINT IN X-Y PLANE |
| RGAE | RADIUS OF CURVATURE OF EDGE AT DIFFRACTION POINT IN END CAP PLANE |
| SBO | SINE OF BO |
| SPE | SINE OF PHER |
| SPX<br>SPY<br>SPZ | X,Y,Z COMPONENTS OF UNIT VECTOR OF PROPAGATION DIRECTION OF INCIDENT RAY |
| SSBO | SINE OF BO SQUARED |
| STE | SINE OF THER |

| | |
|---|---|
| SV | SINE OF VR |
| T1<br>T2<br>T3 | X,Y,Z COMPONENTS DEFINING THE INCIDENT (OR DIFF) RAY PROPAGATION DIRECTION IN DIFFRACTION POINT COORD SYSTEM |
| THEDR | THETA COMPONENT OF DIFFRACTED RAY DIRECTION IN DIFFRACTION POINT COORDINATE SYSTEM |
| THER | THETA COMPONENT OF INCIDENT RAY PROPAGATION DIRECTION IN DIFFRACTION POINT COORDINATE SYSTEM |
| THEX<br>THEY<br>THEZ | POLARIZATION UNIT VECTOR IN THETA DIRECTION FCR INCIDENT OR DIFFRACTED RAY IN DIFFRACTION POINT COORD SYSTEM IN (X,Y,Z) RCS COMPONENTS |
| THIR | THETA COMPONENT OF INCIDENT RAY DIRECTION IN RCS |
| TOP | COMPUTATIONAL VAPIABLE |
| UB | X,Y,Z COMPONENTS OF UNIT VECTOR TANGENT TO CYLINDER AT DIFFRACTION POINT (2-D) |
| UN | X,Y,Z COMPONENTS OF UNIT NORMAL TO CYLINDER AT DIFFRACTION POINT (2-D) |
| UNEM | NORMALIZATION CONSTANT FOR EDGE UNIT NORMAL NE. |
| UNEX<br>UNEY<br>UNEZ | X,Y,Z COMPONENTS OF UNIT NORMAL TO EDGE IN END CAP PLANE IN RCS |
| V | ELL ANGLES DEFINING (UP TO) 4 DIFFRACTION POINTS ON END CAP NC |
| VR | ELL ANGLE DEFINING DIFFRACTION POINT IN ERCS |
| VXS | X,Y,Z COMPONENTS OF UNIT VECTORS DEFINING SOURCE COORDINATE SYSTEM AXES DIRECTIONS IN RCS |
| XC | X,Y,Z COMPONENTS OF DIFFRACTION POINT LOCATION IN RCS |
| XEX<br>XEY<br>XEZ | X,Y,Z COMPONENTS DEFINING UNIT VECTOR OF X AXIS OF DIFFRACTION POINT COORDINATE SYSTEM (VECTOR NORMAL TO EDGE AND PARALLEL TO END CAP PLANE) |
| YEX<br>YEZ | X AND Z COMPONENTS DEFINING UNIT VECTOR OF Y AXIS OF DIFF. POINT CCORD SYS (VECTOR NORMAL TO END CAP) |

CODE LISTING

```
C--------------------------------------------------------------------
      SUBROUTINE ENDIF(EDTH,EDPH,NC)
C!!!
C!!!  COMPUTES THE DIFFRACTED FIELD FROM THE END CAP RIM
C!!!
      COMPLEX EDTH,EDPH,EIX,EIY,EIZ,EIPR,EIPP,PH,EDPR,EDPP,DS,DH
      COMPLEX CJ,CPI4,EF,EG
      DIMENSION V(4),UN(2),UB(2),DI(3),XC(3)
      LOGICAL LHIT,LDEBUG,LTEST
      COMMON/DIR/D(3),THSR,PHSR,SPS,CPS,STHS,CTHS
      COMMON/GEOMEL/A,B,ZC(2),SNC(2),CNC(2),CTC(2)
      COMMON/SORINF/XS(3),VXS(3,3)
      COMMON/COMP/CJ,CPI4
      COMMON/PIS/PI,TPI,DPR,RPD
      COMMON/THPHUV/DT(3),DP(2)
      COMMON/TEST/LDEBUG,LTEST
      EDTH=(Ø.,Ø.)
      EDPH=(Ø.,Ø.)
      IF(LDEBUG) WRITE(6,9ØØ)
C!!!  CALCULATE DIFFRACTION POINTS
9ØØ   FORMAT(/,' DEBUGGING ENDIF SUBROUTINE')
      CALL DFPTCL(V,NC)
      IF(LDEBUG) WRITE(6,*) NC,V
C!!!  STEP THRU DIFFRACTION POINTS
      DO 1 I=1,4
      IF(V(I).LT.-5ØØ.) GO TO 2
C!!!  SET UP INCIDENT RAY GEOMETRY
      VR=V(I)*RPD
      SV=SIN(VR)
      CV=COS(VR)
      XC(1)=A*CV
      XC(2)=B*SV
      XC(3)=A*CTC(NC)*CV+ZC(NC)
C!!!  DOES DIFFRACTED RAY HIT A PLATE?
      CALL PLAINT(XC,D,DHIT,Ø,LHIT)
      IF(LHIT) GO TO 1
      SPX=XC(1)-XS(1)
      SPY=XC(2)-XS(2)
      SPZ=XC(3)-XS(3)
      SPM=SQRT(SPX*SPX+SPY*SPY+SPZ*SPZ)
      SPX=SPX/SPM
      SPY=SPY/SPM
      SPZ=SPZ/SPM
      TOP=SQRT(SPX*SPX+SPY*SPY)
      THIR=BTAN2(TOP,SPZ)
      PHIR=BTAN2(SPY,SPX)
      DI(1)=SPX
      DI(2)=SPY
      DI(3)=SPZ
C!!!  DOES INCIDENT RAY HIT PLATE BEFORE END CAP?
      CALL PLAINT(XS,DI,DHIT,Ø,LHIT)
      IF(LHIT.AND.(DHIT.LT.SPM)) GO TO 1
C!!!  CALCULATE INCIDENT FIELD PATTERN FACTOR
      CALL SOURCE(EF,EG,EIX,EIY,EIZ,THIR,PHIR,VXS)
      IF(LDEBUG) WRITE(6,*) EF,EG
      EX=-A*SV
      EY=B*CV
      EZ=-A*CTC(NC)*SV
      EM=SQRT(EX*EX+EY*EY+EZ*EZ)
      NCC=NC
      IF(NCC.GT.1)NCC=-1
C!!!  CALCULATE DIF. POINT COORD. SYS UNIT AXES AND RELATED GEOM.
      EX=NCC*EX/EM
      EY=NCC*EY/EM
      EZ=NCC*EZ/EM
      CBO=D(1)*EX+D(2)*EY+D(3)*EZ
      IF(CBO.GT.1.) CBO=1.
      SBO=SQRT(1.-CBO*CBO)
      SSBO=SBO*SBO
      UNEX=B*CV*SNC(NC)
      UNEY=A*SV/SNC(NC)
      UNEZ=B*CNC(NC)*CV
      UNEM=SQRT(UNEX*UNEX+UNEY*UNEY+UNEZ*UNEZ)
      UNEX=UNEX/UNEM
      UNEY=UNEY/UNEM
      UNEZ=UNEZ/UNEM
      RG=((A*A*SV*SV+B*B*CV*CV)**(1.5))/A/B
      RGAE=A*A*SV*SV+B*B*SNC(NC)*SNC(NC)*CV*CV
      RGAE=(RGAE**1.5)/A/B
      AE=RGAE/SNC(NC)/SNC(NC)
      CALL NANDB(UN,UB,VR)
      YEX=-CNC(NC)*NCC
      YEZ=SNC(NC)*NCC
      XEX=-YEZ*EY
      XEY=YEZ*EX-YEX*EZ
      XEZ=YEX*EY
      T1=XEX*SPX+XEY*SPY+XEZ*SPZ
      T2=YEX*SPX+YEZ*SPZ
      T3=EX*SPX+EY*SPY+EZ*SPZ
      THER=BTAN2(SQRT(T1*T1+T2*T2),-T3)
      PHER=BTAN2(-T2,-T1)
      IF(PHER.LT.0.) PHER=TPI+PHER
      FN=1.+ACOS(UN(1)*YEX)/PI
      IF(PHER.GT.FN*PI)GO TO 1
      CTE=COS(THER)
      STE=SIN(THER)
      CPE=COS(PHER)
      SPE=SIN(PHER)
C111  CALCULATE INCIDENT FIELD THETA AND PHI POLARIZATION
C111  UNIT VECTORS
      THEX=XEX*CTE*CPE+YEX*CTE*SPE-EX*STE
      THEY=XEY*CTE*CPE-EY*STE
      THEZ=XEZ*CTE*CPE+YEZ*CTE*SPE-EZ*STE
      PHEX=-XEX*SPE+YEX*CPE
      PHEY=-XEY*SPE
      PHEZ=-XEZ*SPE+YEZ*CPE
C111  COMPUTE COMPONENTS OF INC. FIELD PERPENDICULAR AND PARALLEL
C111  TO THE EDGE
      EIPR=EIX*PHEX+EIY*PHEY+EIZ*PHEZ
      EIPP=EIX*THEX+EIY*THEY+EIZ*THEZ
C111  COMPUTE PARAMETERS USED IN DIF. COEF. CALCULATIONS
      T1=UNEX*(SPX-D(1))+UNEY*(SPY-D(2))+UNEZ*(SPZ-D(3))
      R=SPM*AE*SSBO/(AE*SSBO-T1*SPM)
      FLI=SPM*SSBO
      FLRO=SPM*SSBO
      T1=UN(1)*UNEX+UN(2)*UNEY
      CTHI=-(SPX*UN(1)+SPY*UN(2))
      RRN=SPM*AE*SSBO/(AE*SSBO+2*T1*CTHI*SPM)
      WR=BTAN2(-SPX*UB(1)-SPY*UB(2),-SPZ)
      SSW=SIN(WR)**2
      SCW=COS(WR)**2
      SST2=SSW+SCW*CTHI*CTHI
      RHO2=SPM
      RHO1=SPM*RG*CTHI/(RG*CTHI+2.*SPM*SST2)
      IF(CTHI.LT.1.E-5)RHO1=SPM
      FLRN=RHO1*RHO2*SSBO/RRN
      T1=XEX*D(1)+XEY*D(2)+XEZ*D(3)
      T2=YEX*D(1)+YEZ*D(3)
      T3=EX*D(1)+EY*D(2)+EZ*D(3)
      THEDR=BTAN2(SQRT(T1*T1+T2*T2),T3)
      PHEDR=BTAN2(T2,T1)
      IF(PHEDR.LT.0.) PHEDR=TPI+PHEDR
      IF(PHEDR.GT.FN*PI)GO TO 1
      CTE=COS(THEDR)
      STE=SIN(THEDR)
      CPE=COS(PHEDR)
      SPE=SIN(PHEDR)
C!!!  CALCULATE PHASE TERM
      PH=CEXP(-CJ*TPI*SPM)/SPM
      PH=PH*CEXP(CJ*TPI*(XC(1)*D(1)+XC(2)*D(2)+XC(3)*D(3)))
C!!!  CALCULATE DIFFRACTION COEFFICIENTS
      CALL DZ(DS,DH,FLI,FLRN,FLRO,PHEDR,PHER,SBO,FN)
      IF(LDEBUG) WRITE(6,*) FLI,FLRN,FLRO,PHEDR,PHER,SBO,FN
      IF(LDEBUG) WRITE(6,*) DS,DH
      IF(R.GE.0.) GO TO 5
      R=ABS(R)
      PH=(0.,1.)*PH
5     CONTINUE
C!!!  CALCULATE DIF. FIELD COMPONENTS PERPENDICULAR AND PARALLEL
C!!!  TO THE EDGE
      EDPR=-DH*SQRT(R)*EIPR*PH
      EDPP=-DS*SQRT(R)*EIPP*PH
C!!!  CALCULATE DIF. FIELD THETA AND PHI POLARIZATION UNIT VECTORS
      THEX=XEX*CTE*CPE+YEX*CTE*SPE-EX*STE
      THEY=XEY*CTE*CPE-EY*STE
      THEZ=XEZ*CTE*CPE+YEZ*CTE*SPE-EZ*STE
      THEX=-THEX
      THEY=-THEY
      THEZ=-THEZ
      PHEX=-XEX*SPE+YEX*CPE
      PHEY=-XEY*SPE
      PHEZ=-XEZ*SPE+YEZ*CPE
C!!!  CALCULATE THETA AND PHI COMPONENTS OF DIF. FIELD IN RCS
      EDTH=EDTH+EDPR*(PHEX*DT(1)+PHEY*DT(2)+PHEZ*DT(3))
      EDTH=EDTH+EDPP*(THEX*DT(1)+THEY*DT(2)+THEZ*DT(3))
      EDPH=EDPH+EDPR*(PHEX*DP(1)+PHEY*DP(2))
      EDPH=EDPH+EDPP*(THEX*DP(1)+THEY*DP(2))
1     CONTINUE
2     IF(.NOT.LTEST) RETURN
      WRITE(6,910)
910   FORMAT(/,' TESTING ENDIF SUBROUTINE')
      WRITE(6,*) EDTH,EDPH,NC
      RETURN
      END
```

## FCT

### PURPOSE

This function computes the integrand for various integrals used to compute the diffraction coefficient for an elliptic cylinder.

### METHOD

For the present code, only the integrand defined for ID equal to three is used. This is used to define the arc length between two points on the elliptic cylinder. The arc length is given by

$$t = \frac{1}{|\sin\alpha_s|} \int_{v_i}^{v_f} FCT(v)dv \quad ,$$

where

$$FCT(x) = \sqrt{A^2\sin^2x + B^2\cos^2x} \quad .$$

### SYMBOL DICTIONARY

| | |
|---|---|
| A2 | THE SQUARE OF THE RADIUS OF THE ELLIPTIC CYLINDER ON THE X-AXIS |
| B2 | THE SQUARE OF THE RADIUS OF THE ELLIPTIC CYLINDER ON THE Y-AXIS |
| CN | COSINE OF X |
| F | SQRT((A*SIN(VR)**2+(B*COS(VR))**2) |
| SN | SINE OF X |
| SNA | THE ABSOLUTE VALUE OF THE SINE OF THE ANGLE MEASURED FROM THE NEGATIVE Z-AXIS OF THE CYLINDER TO THE DIRECTION OF PROPAGATION |
| X | THE ARGUMENT OF THE INTEGRAND DEFINING THE ELLIPTIC ANGLE |

## CODE LISTING

```
C-------------------------------------------------------------------
      FUNCTION FCT(X)
C!!!
C!!!  THESE ARE INTEGRAND OF ATTENUATION COEFFICIENT INTEGRATION.
C!!!
      COMMON/GEOMEL/A,B,ZC(2),SNC(2),CNC(2),CTC(2)
      COMMON/PIS/PI,TPI,DPR,RPD
      COMMON/GID/AS,ID,SAS,SASP,CAS
      A2 = A*A
      B2 = B*B
      SNA=ABS(SAS)
      SN = SIN(X)
      CS = COS(X)
      SN2 = SIN (2.*X)
      CS2 = COS(2.*X)
      Q = (A2*B2*SNA)**(1./3.)
      GIN = 3.*(A2-B2)/Q
      F = SQRT (A2*SN*SN+B2*CS*CS)
      IF (ID .EQ. 3)   GO TO 3
      IF (ID .EQ. 2)   GO TO 2
      IF (ID .EQ. 4)  GO TO 4
      IF (ID .EQ. 5) GO TO 5
      FCT = Q/F
      RETURN
    2 FCT = Q*C*SNA/(F*F*F)
      RETURN
    3 FCT = F
      RETURN
    4 FCT = GIN*CS2/F
      RETURN
    5 FCT = .75*(A2-B2)*GIN*SN2*SN2/F/F/F
      RETURN
      END
```

## FFCT

### PURPOSE

The purpose of this function is to determine the transition function for the edge and corner diffraction coefficients.

### METHOD

The transition function for the edge and corner diffraction coefficients is given by[4]:

$$FFCT(x) = 2j\left|\sqrt{x}\right| e^{jx} \int_{|\sqrt{x}|}^{\infty} e^{-j\tau^2}\, d\tau.$$

This can also be written as

$$FFCT(x) = j\sqrt{2\pi|x|}\, e^{jx}\left[(0.5-j0.5)-\left(C\left(\sqrt{\frac{2|x|}{\pi}}\right)-jS\left(\sqrt{\frac{2|x|}{\pi}}\right)\right)\right]$$

where

$$\int_0^{\alpha} e^{-j\frac{\pi}{2}t^2}\, dt = C(\alpha) - jS(\alpha).$$

### SYMBOL DICTIONARY

| | |
|---|---|
| CFR | REAL PART OF FRESNEL INTEGRAL |
| DEL | ARGUMENT OF TRANSITION FUNCTION |
| FFCT | TRANSITION FUNCTION |
| S | ARGUMENT OF FRESNEL INTEGRAL |
| SDEL | SQRT(ABS(DEL)) |
| SFR | IMAGINARY PART OF FRESNEL INTEGRAL |

### CODE LISTING

```
C-----------------------------------------------------------------------
      COMPLEX FUNCTION FFCT(DEL)
C!!!
C!!!  DETERMINES THE TRANSITION FUNCTION RESULT FOR THE EDGE
C!!!  AND CORNER DIFFRACTION COEFFICIENTS.
C!!!
      COMMON/PIS/PI,TPI,DPR,RPD
      IF(ABS(DEL).GT.10.) GO TO 1
      SDEL=SQRT(ABS(DEL))
      S=SQRT(2./PI)*SDEL
      CALL FRNELS(CFR,SFR,S)
      FFCT=CMPLX(0.5-CFR,SFR-0.5)
      FFCT=SQRT(TPI)*SDEL*FFCT*CEXP(CMPLX(0.,DEL+PI/2.))
      RETURN
    1 FFCT=(1.,0.)
      RETURN
      END
```

## FKARG

### PURPOSE

To compute a parameter needed in the diffraction coefficient for the elliptic cylinder.

### METHOD

This subroutine computes the parameter used in the diffraction coefficient to determine the fields scattered from the elliptic cylinder. This parameter is given by [6],

$$\xi = \int_{Q_1}^{Q_2} \pi^{1/3} \rho_g^{-2/3} \, dt,$$

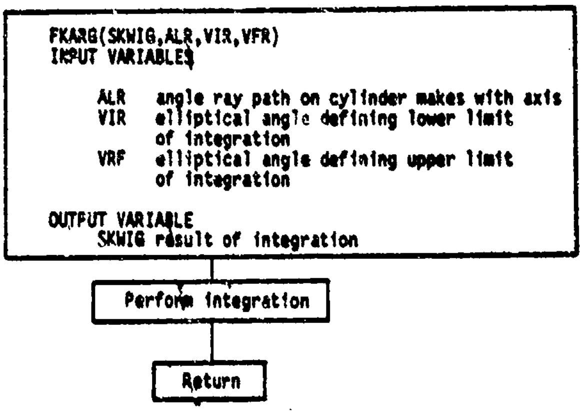
where $\rho_g$ is the radius of curvature of the elliptic cylinder in the plane of propagation. This can also be written as

$$\xi = \pi^{1/2}(AB)^{2/3}|\sin\alpha|^{1/3} \int_{v_i}^{v_f} \frac{dv}{\sqrt{A^2\sin^2 v + B^2\cos^2 v}} \quad ,$$

where

$\xi$ = SKWIG
$\alpha$ = ALR
$v_i$ = VIR
$v_f$ = VFR.

### FLOW DIAGRAM


FKARG(SKWIG,ALR,VIR,VFR)
INPUT VARIABLES
ALR angle ray path on cylinder makes with axis
VIR elliptical angle defining lower limit of integration
VRF elliptical angle defining upper limit of integration
OUTPUT VARIABLE
SKWIG result of integration
Perform integration
Return

## SYMBOL DICTIONARY

ALR ANGLE MEASURED FROM NEGATIVE Z-AXIS IN THE DIRECTION OF PROPAGATION

ANS THE EVALUATED INTEGRAL

FUNI INTEGRAND OF THE INTEGRAL

SKWIG PARAMETER USED TO DEFINE CURVED SURFACE AT THE POINT OF DIFFRACTION

VFR ELLIPTICAL ANGLE DEFINING THE DIFFRACTION ANGLE POSITION ON THE CYLINDER

VIR ELLIPTICAL ANGLE DEFINING THE INCIDENT ANGLE POSITION ON CYLINDER

## CODE LISTING

```
C--------------------------------------------------------------------
      SUBROUTINE FKARG(SKWIG,ALR,VIR,VFR)
C!!!
C!!!  COMPUTES THE PARAMETER NEEDED IN THE DIFFRACTION
C!!!  COEFFICIENT FOR THE ELLIPTIC CYLINDER
C!!!
      COMMON/PIS/PI,TPI,DPR,RPD
      COMMON/GEOMEL/A,B,ZC(2),SNC(2),CNC(2),CTC(2)
      EXTERNAL FUNI
      IF(ABS(VIR-VFR).LT.1.E-5)GO TO 1
      SKWIG=(PI*ABS(SIN(ALR)))**(1./3.)
      SKWIG=SKWIG*((A*B)**(2./3.))
      CALL DQG32(VIR,VFR,FUNI,ANS)
      SKWIG=SKWIG*ANS
      SKWIG=ABS(SKWIG)
      RETURN
1     SKWIG=0.
      RETURN
      END
```

## FKY

### PURPOSE

This function is used in computing the transition function for curved edge diffraction.

### METHOD

The transition function for the diffraction coefficient of an edge in a curved surface is the same as for a straight wedge, except that the curved edge function takes into account the possibility of the distance parameter being negative. The transition function is given by [4 ]

$$F(x) = 2j|\sqrt{x}|e^{jx} \int_{|\sqrt{x}|}^{\infty} e^{-j\tau^2} d\tau,$$

where $x = kLa,$

and $k = 2\pi/\lambda$

$L$ = distance parameter

$a$ = a function dependent on the square of the cosine of the incident and diffraction angles and the wedge angle number.

The transition function can then be written as,

$$F(kLa) = j2\pi\sqrt{\frac{|L|a}{\lambda}}\; e^{jk|L|a}\left[(0.5-j0.5) - \left(C\left(2\sqrt{\frac{|L|a}{\lambda}}\right) - jS\left(2\sqrt{\frac{|L|a}{\lambda}}\right)\right)\right], \quad \text{for } L>0$$

and

$$F(kLa) = F^*(k|L|a), \qquad \text{for } L<0$$

where the "*" means the complex conjugate and

$$\int_0^{\alpha} e^{-j\frac{\pi}{2}t^2} dt = C(\alpha) - jS(\alpha).$$

The above equation relates to the function FKY as,

$$FKY(L/\lambda, a) = F(kLa).$$

## SYMBOL DICTIONARY

| | |
|---|---|
| A | PARAMETER DEPENDANT ON THE INCIDENT AND DIFFRACTED ANGLES |
| C | REAL PART OF FRESNEL INTEGRAL |
| FKY | TRANSITION FUNCTION |
| FL | THE DISTANCE PARAMETER IN WAVELENGTHS |
| FLA | ABSOLUTE VALUE OF FL |
| S | IMAGINARY PART OF FRESNEL INTEGRAL |
| XS | ARGUMENT OF FRESNEL INTEGRAL |

## CODE LISTING

```
C-----------------------------------------------------------------------
      FUNCTION FKY(FL,A)
C!!!
C!!!  TRANSITION FUNCTION FOR CURVED EDGE DIFFRACTION
C!!!
      COMPLEX FKY
      COMMON/PIS/PI,TPI,DPR,RPD
      FLA=ABS(FL)
      XS=2.*SQRT(FLA*A)
      FKY=CMPLX(Ø.,TPI)*SQRT(FLA*A)
      FKY=FKY*CEXP(CMPLX(Ø.,TPI*FLA*A))
      CALL FRNELS(C,S,XS)
      FKY=FKY*CMPLX(.5-C,S-.5)
      IF(FL.GE.Ø.) RETURN
      FKY=CONJG(FKY)
      RETURN
      END
```

## FRNELS

### PURPOSE

To compute the Fresnel integral,

$$f(x_s) = \int_0^{x_s} e^{-j\pi/2\ u^2}\, du = C(x_s) - j\ S(x_s)\ .$$

### METHOD

The integral is evaluated using an approximation by J. Boersma [13]. The integral

$$f(x) = \int_0^x \frac{e^{-jt}}{\sqrt{2\pi t}}\, dt$$

is approximated as follows:

$$\text{for } 0 \le x \le 4 \quad f(x) \simeq e^{-jx} \sqrt{\frac{x}{4}} \sum_{n=0}^{11} (a_n + jb_n) \left(\frac{x}{4}\right)^n$$

$$\text{for } x \ge 4 \quad f(x) \simeq \frac{1-j}{2} + e^{-jx} \sqrt{\frac{4}{x}} \sum_{n=0}^{11} (c_n + jd_n) \left(\frac{4}{x}\right)^n$$

(the constants $a_n$, $b_n$, $c_n$ and $d_n$ are provided by Boersma and are defined in data statements in the subroutine).

Note that by performing a change of variable, the integral to be solved becomes of the form of the integral which Boersma solved;

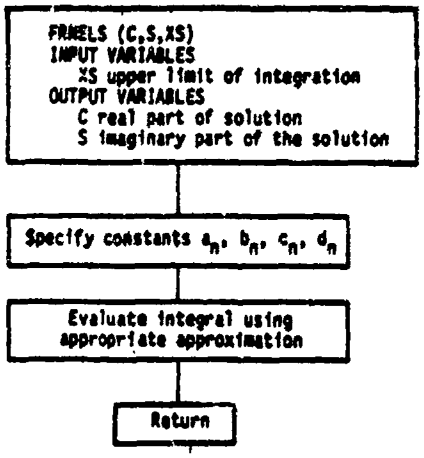
$$t = \frac{\pi}{2}\ u^2\ .$$

By applying this change of variable, we get

$$f(x_s) = \int_0^{x_s} e^{-j\frac{\pi}{2}u^2}\, du = \int_0^x \frac{e^{-jt}}{\sqrt{2\pi t}}\, dt$$

where $\quad x = \frac{\pi}{2} x_s^2.$

## FLOW DIAGRAM


FRNELS (C,S,XS)
INPUT VARIABLES
XS upper limit of integration
OUTPUT VARIABLES
C real part of solution
S imaginary part of the solution
Specify constants $a_n$, $b_n$, $c_n$, $d_n$
Evaluate integral using appropriate approximation
Return

## SYMBOL DICTIONARY

| | |
|---|---|
| A<br>B<br>C<br>D | CONSTANTS USED IN EVALUATING INTEGRAL |
| FI | IMAGINARY COMPONENT OF SUMMATION FUNCTION |
| FR | REAL COMPONENT OF SUMMATION FUNCTION |

## CODE LISTING

```
C-----------------------------------------------------------------------
      SUBROUTINE FRNELS(C,S,XS)
C!!!
C!!!  THIS IS THE FRESNEL INTEGRAL SUBROUTINE WHERE THE INTEGRAL
C!!!  IS FROM U=0 TO XS, THE INTEGRAND IS EXP(-J*PI/2.*U*U),
C!!!  AND THE OUTPUT IS C(XS)-J*S(XS).
C!!!
      LOGICAL LDEBUG,LTEST
      COMMON/TEST/LDEBUG,LTEST
      COMMON/PIS/PI,TPI,DPR,RPD
      DIMENSION A(12),B(12),CC(12),D(12)
C!!!  SPECIFY CONSTANTS
      DATA A/1.595769140,-0.000001702,-6.808568854,-0.000576361,6.92069
     1
     2902,-0.016898657,-3.050485660,-0.075752419,0.850663781,-0.02
563904
     21,-0.150230960,0.034404779/
      DATA B/-0.000000033,4.255387524,-0.000092810,-7.780020400,-0.0095
     2
     20895,5.075161298,-0.138341947,-1.363729124,-0.403349276,0.70
222201
     26,-0.216195929,0.019547031/
      DATA CC/0.,-0.024933975,0.000003936,0.005770956,0.000689892,-0.00
     9
     2497136,0.011948809,-0.006748873,0.000246420,0.002102967,-0.0
012179
     230,0.000233939/
      DATA D/0.199471140,0.000000023,-0.009351341,0.000023006,0.0048514
     6
     26,0.001903218,-0.017122914,0.029064067,-0.027928955,0.016497
308,-0
     2.005598515,0.000838386/
      IF(XS.LE.0.0) GO TO 414
      X=XS
      X = PI*X*X/2.0
      FR=0.0
      FI=0.0
      K=13
C!!!  IS X<4?
      IF(X-4.0) 10,40,40
 10   Y=X/4.0
C!!!  EVALUATE INTEGRAL USING X<4 APPROXIMATION
 20   K=K-1
      FR=(FR+A(K))*Y
      FI=(FI+B(K))*Y
      IF(K-2) 30,30,20
 30   FR=FR+A(1)
      FI=FI+B(1)
      C=(FR*COS(X)+FI*SIN(X))*SQRT(Y)
      S=(FR*SIN(X)-FI*COS(X))*SQRT(Y)
      GO TO 1
C!!!  EVALUATE INTEGRAL USING X>4 APPROXIMATION
 40   Y=4.0/X
 50   K=K-1
      FR=(FR+CC(K))*Y
      FI=(FI+D(K))*Y
      IF(K-2) 60,60,50
 60   FR=FR+CC(1)
      FI=FI+D(1)
      C=0.5+(FR*COS(X)+FI*SIN(X))*SQRT(Y)
      S=0.5+(FR*SIN(X)-FI*COS(X))*SQRT(Y)
      GO TO 1
 414  C=-0.0
      S=-0.0
 1    IF (.NOT.LTEST) GO TO 2
```

```
         WRITE (6,3)
    3    FORMAT (/,' TESTING FRNELS SUBROUTINE')
         WRITE (6,*) C,S,XS
    2    RETURN
         END
```

## FUNI

### PURPOSE

This function calculates the integrand of the integral in subroutine FKARG.

### METHOD

The integrand of this integral evaluated in subroutine FKARG is given by

$$\mathrm{FUNI(VR)} = \frac{1}{\sqrt{A^2\sin^2(\mathrm{VR})+B^2\cos^2(\mathrm{VR})}} \quad .$$

### SYMBOL DICTIONARY

| | |
|---|---|
| A | RADIUS OF CYLINDER ON X-AXIS |
| B | RADIUS OF CYLINDER ON Y-AXIS |
| VR | ELLIPTIC ANGLE ON CYLINDER IN RADIANS |

### CODE LISTING

```
C-----------------------------------------------------------------
      FUNCTION FUNI(VR)
C!!!
C!!!  INTEGRAND OF INTEGRAL NEEDED IN FKARG
C!!!
      COMMON/GEOMEL/A,B,ZC(2),SNC(2),CNC(2),CTC(2)
      FUNI=1./SQRT(A*A*SIN(VR)*SIN(VR)+B*B*COS(VR)*COS(VR))
      RETURN
      END
```

## GEOM

PURPOSE

This subroutine calculates a large number of constants that are fixed for a given geometry of plates. They are stored in common blocks for use in other sections of the program. It is called once for every source used. Because of the diversity of operations done in GEOM, it's description is broken into seven parts:

1. Identify edges which are common to more than one plate.
2. Compute unit vectors of edge-fixed coordinate systems for each edge on each plate.
3. Determine source image information for reflections from plates.
4. Calculate possible range for diffraction angle $\beta_o$ for each edge.
5. Determine wedge angles for plates with common edges.
6. Determine plates which are totally shadowed from the source.
7. Perform calculations for plates which intersect.

GEOM, SECTION 1

PURPOSE

To identify edges which are common to two plates.

PERTINENT GEOMETRY

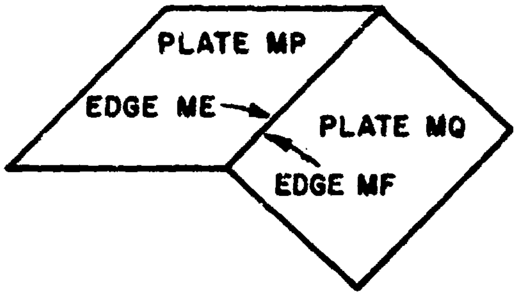
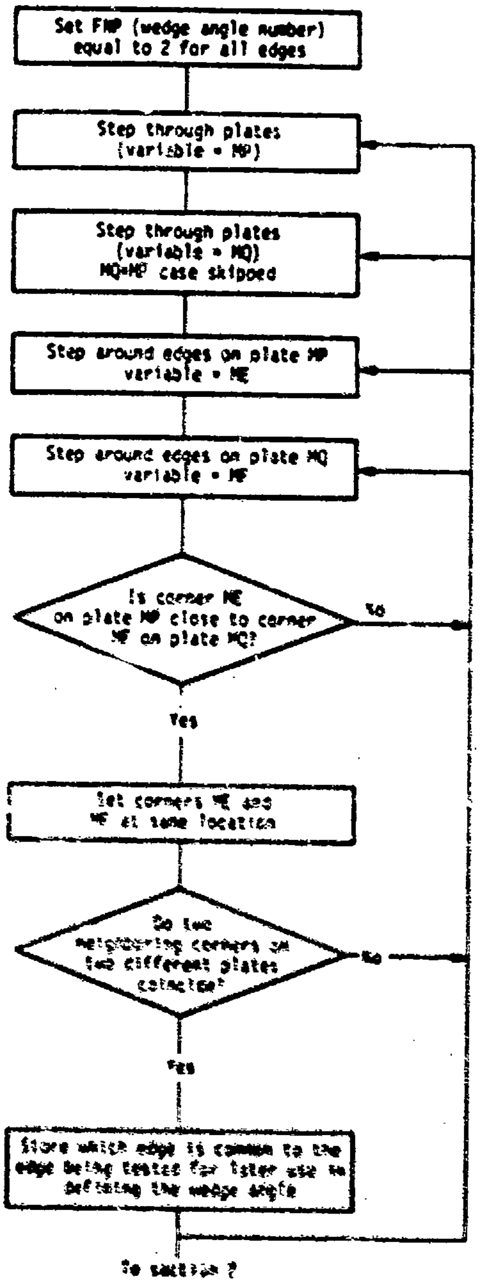


Figure 62--Illustration of two plates with a common edge.

## FLOW DIAGRAM


Set FNP (wedge angle number) equal to 2 for all edges
Step through plates (variable = MP)
Step through plates (variable = MQ) MQ=MP case skipped
Step around edges on plate MP variable = ME
Step around edges on plate MQ variable = MF
Is corner ME on plate MP close to corner MF on plate MQ?
No
Yes
Set corners ME and MF at same location
Do two neighboring corners on two different plates coincide?
No
Yes
Store which edge is common to the edge being tested for later use in defining the wedge angle
To section 2

## CODE LISTING

```
C-----------------------------------------------------------------
      SUBROUTINE GEOM
C!!!
C!!!  THIS ROUTINE COMPUTES ALL THE GEOMETRY ASSOCIATED
C!!!  WITH FIXED PLATE STRUCTURE,SUCH AS EDGE UNIT VECTORS,
C!!!  PLATE NORMALS,SHADOWED PLATES,ETC.
C!!!
      DIMENSION IHIT(6),XII(3),XIN(3),VI(3),DS(3),XC(3),XSI(3),XSII(3)
      DIMENSION XOB(3),XDC(3),VTCP(2),BTCP(4),VTCN(2),BTCN(4)
      DIMENSION VAX(3,3)
      LOGICAL LSURF,LNPL
      LOGICAL LSHD,LSTD,LSTS,LCTD,LHCT,LHIT
      LOGICAL LGRND,LIND,LDEBUG,LTEST
      COMMON/TEST/LDEBUG,LTEST
      COMMON/PIS/PI,TPI,DPR,RPD
      COMMON/GEOPLA/X(14,6,3),V(14,6,3),VP(14,6,3),VN(14,3)
     2,NEP(14),NPX
      COMMON/EDMAG/VMAG(14,6)
      COMMON/SCRINF/XS(3),VXS(3,3)
      COMMON/IMAINF/XI(14,14,3),VXI(3,3,14)
      COMMON/BNDFCL/BD(14,6,2)
      COMMON/SURFAC/LSURF(14)
      COMMON/LSHDT/LSHD(14),LIND(14,14)
      COMMON/LSHDP/LSTS,LSTD(14)
      COMMON/FNANG/FNP(14,6)
      COMMON/HITPLT/NPH
      COMMON/SOURSF/FACTOR
      COMMON/FARP/IP,H,HAS
      COMMON/GROUND/LGRND,NPXN
      LSTS=.FALSE.
C!!!  SECTION 1 ********************************
C!!!  DETERMINATION OF COMMON EDGES
C!!!  SET FNP=2 FOR ALL EDGES
      DO 3 NP=1,NPXN
      NEX=NEP(NP)
      DO 3 NE=1,NEX
3     FNP(NP,NE)=2.
C!!!  STEP THROUGH PLATES (PLATE NP)
      DO 17 NP=1,NPX
      NEX=NEP(NP)
C!!!  STEP THROUGH PLATES (PLATE NO, WHERE NO.NE.NP)
      DO 16 NO=1,NPX
      IF(NO.EQ.NP) GO TO 16
      NFX=NEP(NO)
      NEC=0
      NFC=0
C!!!  STEP AROUND EDGES ON PLATE NP
      DO 12 NE=1,NEX
C!!!  STEP AROUND EDGES ON PLATE NO
      DO 5 NF=1,NFX
      XN=0.
C!!!  IS CORNER NE ON PLATE NP CLOSE TO CORNER NF ON PLATE NO?
      DO 4 N=1,3
4     XN=XN+(X(NO,NF,N)-X(NP,NE,N))*(X(NO,NF,N)-X(NP,NE,N))
      IF(XN.LT.0.01) GO TO 6
5     CONTINUE
      GO TO 12
6     CONTINUE
      DO 7 N=1,3
C!!!  IF EDGES ARE CLOSE, SET THEM IDENTICAL
7     X(NO,NF,N)=X(NP,NE,N)
      IF(NEC.NE.0) GO TO 8
C!!!  CHECK TO SEE IF TWO NEIGHBORING CORNERS ON TWO PLATES
C!!!  COINCIDE
      NEC=NE
      MFC=MF
      GO TO 12
8     MES=ME-MEC
      IF(MES.EQ.1.OR.MES+1.EQ.MEX) GO TO 18
      GO TO 12
18    MEN=MEC
      IF(MES+1.EQ.MEX) MEN=MEX
      MFS=IABS(MF-MFC)
      IF(MFS.EQ.1.OR.MFS+1.EQ.MFX) GO TO 19
      GO TO 12
19    MFN=MFC
      IF(MFS+1.EQ.MFX) MFN=MFX
      IF(MF-MFC.EQ.-1) MFN=MF
      IF(FNP(MP,MEN).GT.0.) GO TO 9
      NFN=FNP(MP,MEN)-.5
      NFN=IABS(NFN)/100
      IF(NFN.EQ.MO) GO TO 12
9     CONTINUE
C!!!  STORE WHICH EDGE IS COMMON TO THE EDGE BEING TESTED
C!!!  FOR LATER USE IN DEFINING WEDGE ANGLES
      IF(FNP(MO,MFN).GT.0.) FNP(MO,MFN)=-1.*(100*MP+MEN)
      IF(FNP(MP,MEN).GT.0.) FNP(MP,MEN)=-1.*(100*MO+MFN)
12    CONTINUE
16    CONTINUE
17    CONTINUE
      IF (LDEBUG) WRITE (6,667)
667   FORMAT (/,' DEBUGGING GEOM SUBROUTINE')
```

## GEOM SECTION 2

PURPOSE

This section computes the edge-fixed coordinate system unit vectors for each edge.

PERTINENT GEOMETRY

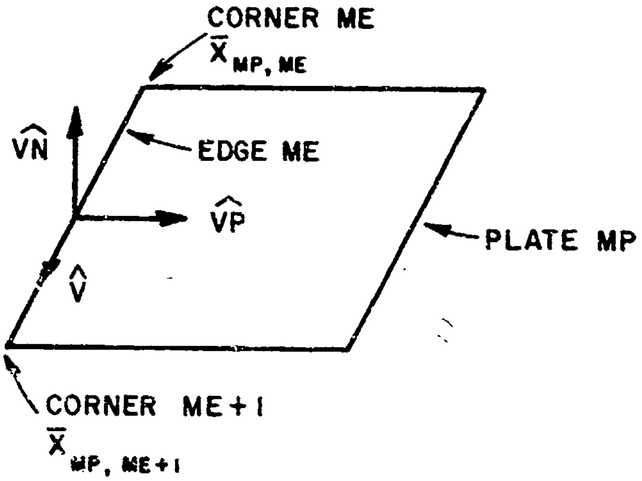


Figure 63--Edge coordinate system unit vectors.

$\hat{V}_{MP,ME}$ = edge unit vector = $\hat{x}$ V(MP,ME,1) + $\hat{y}$ V(MP,ME,2) + $\hat{z}$ V(MP,ME,3)

$\hat{VN}_{MP}$ = plate unit normal = $\hat{x}$ VN(MP,1) + $\hat{y}$ VN(MP,2) + $\hat{z}$ VN(MP,3)

$\hat{VP}_{MP,ME}$ = edge unit binormal = $\hat{x}$ VP(MP,ME,1) + $\hat{y}$ VP(MP,ME,2) + $\hat{z}$ VP(MP,ME,3)

$\overline{X}_{MP,ME}$ = corner location = $\hat{x}$ X(MP,ME,1) + $\hat{y}$ X(MP,ME,2) + $\hat{z}$ X(MP,ME,3)

METHOD

The edge unit vectors are found by,

$$\hat{V}_{MP,ME} = \frac{\overline{X}_{MP,ME+1} - \overline{X}_{MP,ME}}{|\overline{X}_{MP,ME+1} - \overline{X}_{MP,ME}|} .$$

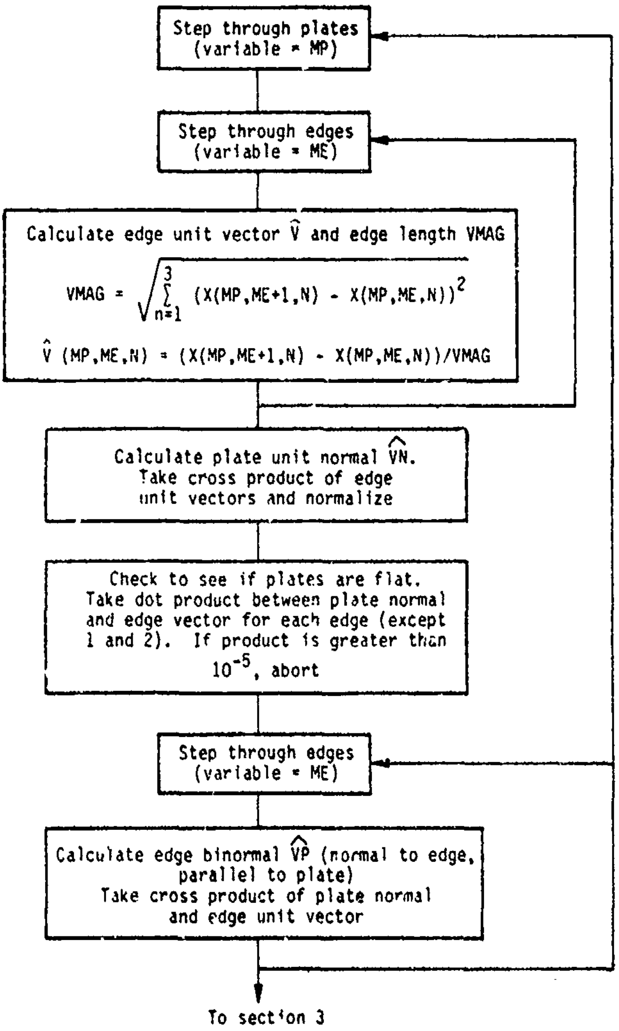
The normals are found using

$$\hat{VN}_{MP} = \frac{\sum_{N=1}^{MEX} \hat{V}_{MP,N} \times \hat{V}_{MP,N+1}}{|\sum_{N=1}^{MEX} \hat{V}_{MP,N} \times \hat{V}_{MP,N+1}|} ,$$

which is an average over the normals computed by all the edges of the plate. This avoids a possible incorrect normal due to a convex edge geometry. The binormals are found by,

$$\hat{VP}_{MP,ME} = \hat{VN}_{MP} \times \hat{V}_{MP,ME}.$$

## FLOW DIAGRAM


Step through plates (variable = MP)
Step through edges (variable = ME)
Calculate edge unit vector $\hat{V}$ and edge length VMAG
$VMAG = \sqrt{\sum_{n=1}^{3} (X(MP,ME+1,N) - X(MP,ME,N))^2}$
$\hat{V}(MP,ME,N) = (X(MP,ME+1,N) - X(MP,ME,N))/VMAG$
Calculate plate unit normal $\widehat{VN}$. Take cross product of edge unit vectors and normalize
Check to see if plates are flat. Take dot product between plate normal and edge vector for each edge (except 1 and 2). If product is greater than $10^{-5}$, abort
Step through edges (variable = ME)
Calculate edge binormal $\widehat{VP}$ (normal to edge, parallel to plate) Take cross product of plate normal and edge unit vector
To section 3

CODE LISTING

```
C!!!  SECTION 2  *******************************
C!!!  DETERMINAION OF V,VN,AND VP UNIT VECTORS FOR EDGE-FIXED
C!!!  COORDINATE SYSTEM
C!!!  STEP THRU PLATES
      DO 1ØØ MP=1,MPXR
      MEX=MEP(MP)
C!!!  STEP THRU EDGES
      DO 15 ME=1,MEX
      MME=ME+1
      IF(MME.GT.MEX) MME=1
      VM=Ø.
C!!!  CALCULATE EDGE UNIT VECTOR V AND EDGE LENGTH VMAG
      DO 1Ø N=1,3
      V(MP,ME,N)=X(MP,MME,N)-X(MP,ME,N)
 1Ø   VM=VM+V(MP,ME,N)*V(MP,ME,N)
      VMAG(MP,ME)=SQRT(VM)
      DO 11 N=1,3
 11   V(MP,ME,N)=V(MP,ME,N)/VMAG(MP,ME)
 15   CONTINUE
      IF(.NOT.LDEBUG)GO TO 991
      DO 992 ME=1,MEX
      WRITE(6,*)(V(MP,ME,N),N=1,3)
 992  CONTINUE
 991  CONTINUE
C!!!  CALCULATE PLATE UNIT NORMAL VN
      VN(MP,1)=Ø.
      VN(MP,2)=Ø.
      VN(MP,3)=Ø.
      DO 22 ME=1,MEX
      MV=ME+1
      IF(MV.GT.MEX) MV=1
      VN(MP,1)=VN(MP,1)+V(MP,ME,2)*V(MP,MV,3)-V(MP,MV,2)*V(MP,ME,3)
      VN(MP,2)=VN(MP,2)+V(MP,ME,3)*V(MP,MV,1)-V(MP,MV,3)*V(MP,ME,1)
      VN(MP,3)=VN(MP,3)+V(MP,ME,1)*V(MP,MV,2)-V(MP,MV,1)*V(MP,ME,2)
 22   CONTINUE
      VNM=Ø.
      DO 2Ø N=1,3
 2Ø   VNM=VNM+VN(MP,N)*VN(MP,N)
      VNM=SQRT(VNM)
      DO 21 N=1,3
 21   VN(MP,N)=VN(MP,N)/VNM
      IF (LDEBUG) WRITE (6,*) (VN(MP,N),N=1,3)
C!!!  INSURE THAT ALL PLATES ARE FLAT. OTHERWISE ABORT!
C!!!  TAKE DOT PRODUCT OF PLATE NORMAL AND EACH EDGE UNIT VECTOR
      DO 12Ø ME=3,MEX
      DOT=VN(MP,1)*V(MP,ME,1)+VN(MP,2)*V(MP,ME,2)+VN(MP,3)*V(MP,ME,3)
      IF(ABS(DOT).LT.1.E-3)GO TO 12Ø
      MEE=ME+1
      WRITE(6,121)MP,MEE
 121  FORMAT(' PLATE # ',I2,' IS NOT FLAT! CORNER # ',I2,' HAS ',
     2'PROBLEM. PROGRAM ABORTS! ******')
      STOP
 12Ø  CONTINUE
C!!!  CALCULATE UNIT BINORMAL VP WHICH IS IN PLATE PLANE
C!!!  AND PERPENDICULAR TO PLATE EDGE
C!!!  TAKE CROSS PRODUCT OF PLATE NORMAL AND EDGE VECTOR
      DO 3Ø ME=1,MEX
      VP(MP,ME,1)=VN(MP,2)*V(MP,ME,3)-VN(MP,3)*V(MP,ME,2)
      VP(MP,ME,2)=VN(MP,3)*V(MP,ME,1)-VN(MP,1)*V(MP,ME,3)
 3Ø   VP(MP,ME,3)=VN(MP,1)*V(MP,ME,2)-VN(MP,2)*V(MP,ME,1)
      IF(.NOT.LDEBUG)GO TO 993
      DO 994 ME=1,MEX
 994  WRITE(6,*)(VP(MP,ME,N),N=1,3)
 993  CONTINUE
 1ØØ  CONTINUE
```

## GEOM SECTION 3

PURPOSE

To calculate source image information for reflection from plates.

PERTINENT GEOMETRY

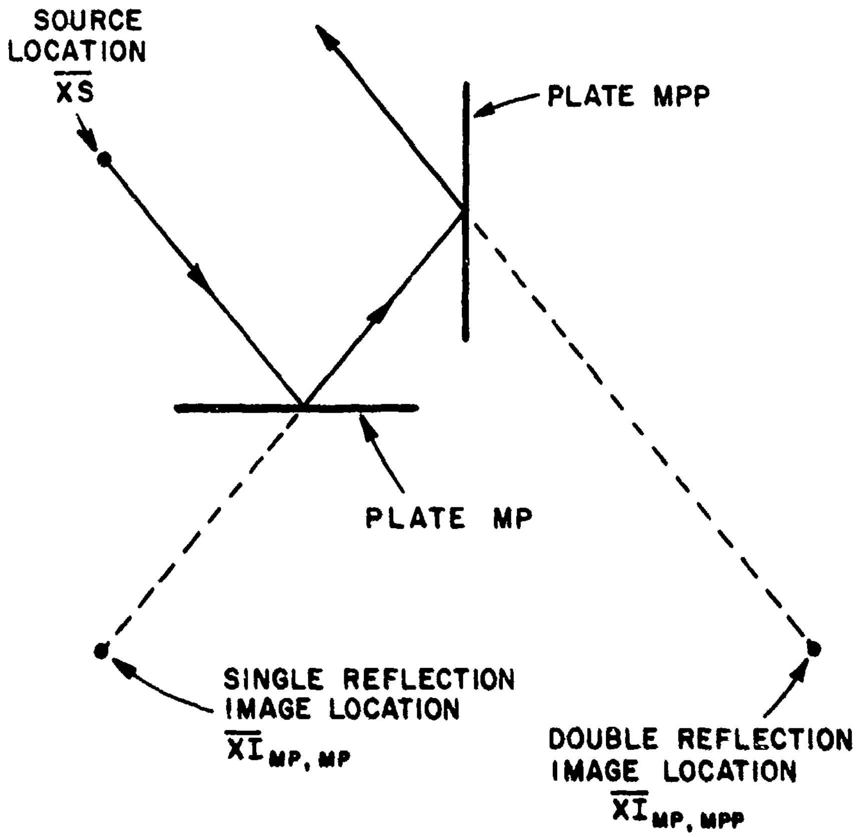
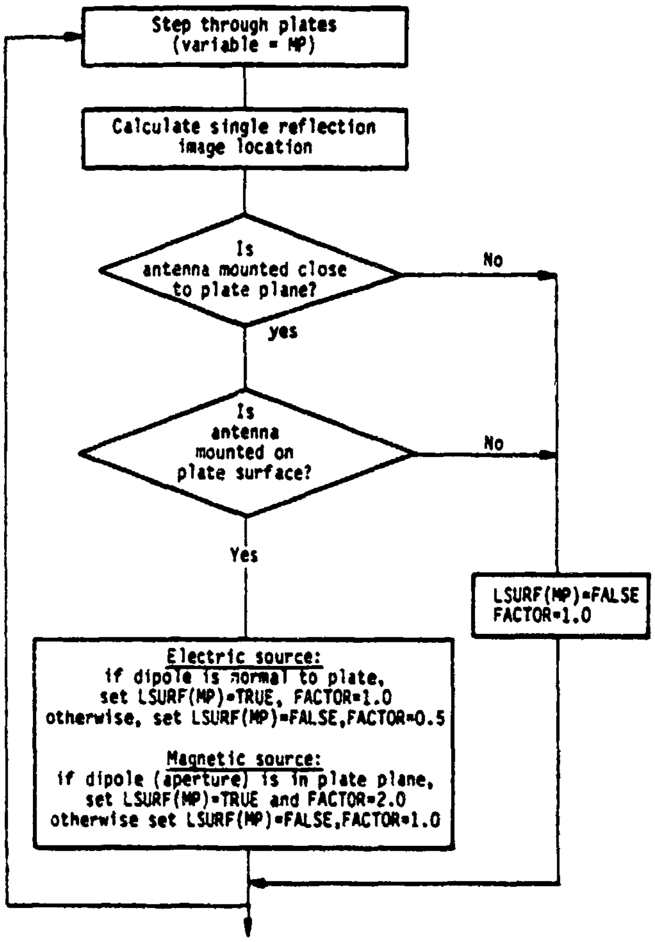


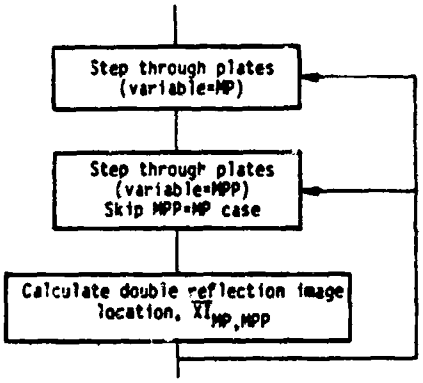
Figure 64--Geometry of image locations for a doubly-reflected ray.

## FLOW DIAGRAM

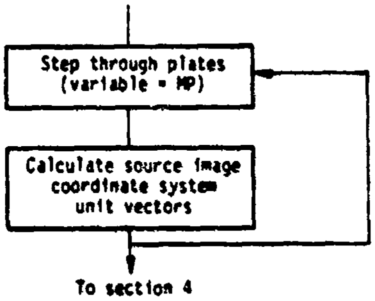
1. Determination of single reflection source image locations and the constant, FACTOR


Step through plates (variable = MP)
Calculate single reflection image location
Is antenna mounted close to plate plane?
No
yes
Is antenna mounted on plate surface?
No
Yes
LSURF(MP)=FALSE
FACTOR=1.0
Electric source:
if dipole is normal to plate,
set LSURF(MP)=TRUE, FACTOR=1.0
otherwise, set LSURF(MP)=FALSE,FACTOR=0.5
Magnetic source:
if dipole (aperture) is in plate plane,
set LSURF(MP)=TRUE and FACTOR=2.0
otherwise set LSURF(MP)=FALSE,FACTOR=1.0

2. Calculate double reflection source image locations



3. Determination of single reflection image dipole directions (image of the source coordinate system axes unit vectors).



```
C!!!    SECTION 3 ********************************
C!!!    DETERMINATION OF SOURCE IMAGE INFORMATION FOR SINGLE
C!!!    AND DOUBLE REFLECTION FROM PLATES
C!!!    1. DETERMINATION OF SINGLE REFLECTION SOURCE IMAGE LOCATIONS
C!!!    AND THE CONSTANT, FACTOR, FOR SOURCES MOUNTED ON THE PLATE
C!!!    SURFACES
        FACTOR=1.
C!!!    STEP THRU PLATES
        DO 50 MP=1,MPXR
        LSURF(MP)=.FALSE.
C!!!    CALCULATE SINGLE REFLECTION IMAGE LOCATION
        CALL IMAGE(XII,XS,AN,MP)
C!!!    IS ANTENNA MOUNTED ON PLATE PLANE?
        IF(ABS(AN).GT.1.E-5)GO TO 560
C!!!    MOVE SOURCE LOCATION SLIGHTLY OFF PLATE IN DIRECTION
C!!!    OF PLATE NORMAL
        DO 566 N=1,3
566     XSI(N)=XS(N)+1.E-5*VN(MP,N)
        CALL IMAGE(XSII,XSI,AN,MP)
        DSM=0.
        DO 563 N=1,3
        DS(N)=XSI(N)-XSII(N)
563     DSM=DSM+DS(N)*DS(N)
        DSM=SQRT(DSM)
        DO 564 N=1,3
        XIN(N)=XSII(N)
564     DS(N)=DS(N)/DSM
        CALL PLAINT(XIN,DS,DHIT,-MP,LHIT)
        IF(.NOT.LHIT)GO TO 560
        DO 567 N=1,3
        XS(N)=XSI(N)
567     XII(N)=XSII(N)
        ENORM=VN(MP,1)*VXS(3,1)+VN(MP,2)*VXS(3,2)+VN(MP,3)*VXS(3,3)
        IF(IN.NE.0)GO TO 561
C!!!    IS MONOPOLE NORMAL TO PLATE?
        IF(1.-ABS(ENORM).GT.1.E-3)GO TO 562
        LSURF(MP)=.TRUE.
        GO TO 560
562     FACTOR=0.5
        GO TO 560
C!!!    IS SLOT IN PLATE PLANE?
561     IF(ABS(ENORM).GT.1.E-3)GO TO 560
        LSURF(MP)=.TRUE.
        FACTOR=2.
560     DO 50 N=1,3
50      XI(MP,MP,N)=XII(N)
C!!!    2. CALCULATE DOUBLE REFLECTION SOURCE IMAGE LOCATIONS
        DO 55 MP=1,MPXR
        DO 55 MPP=1,MPXR
        IF(MP.EQ.MPP) GO TO 55
        DO 51 N=1,3
51      XIN(N)=XI(MP,MP,N)
        CALL IMAGE(XII,XIN,AN,MPP)
        DO 52 N=1,3
52      XI(MP,MPP,N)=XII(N)
55      CONTINUE
        IF (LDEBUG) WRITE (6,*) (((XI(MP,MPP,N),N=1,3),MPP=1,MPXR),
       2MP=1,MPXR)
C!!!    3. DETERMINATION OF SINGLE REFLECTION IMAGE DIPOLE
C!!!    DIRECTIONS
        DO 57 MP=1,MPXR
        CALL IMDIR(VAX,VXS,MP)
        DO 57 NJ=1,3
        DO 57 NI=1,3
57      VXI(NI,NJ,MP)=VAX(NI,NJ)
        IF(.NOT.LDEBUG)GO TO 551
        DO 552 MP=1,MPXR
        DO 552 NI=1,3
552     WRITE(6,*) MP,NI,(VXI(NI,NJ,MP),NJ=1,3)
551     CONTINUE
```

## GEOM SECTION 4

PURPOSE

To determine permissible range for angle $\beta_0$ for diffraction of source ray off of plate edge.

PERTINENT GEOMETRY

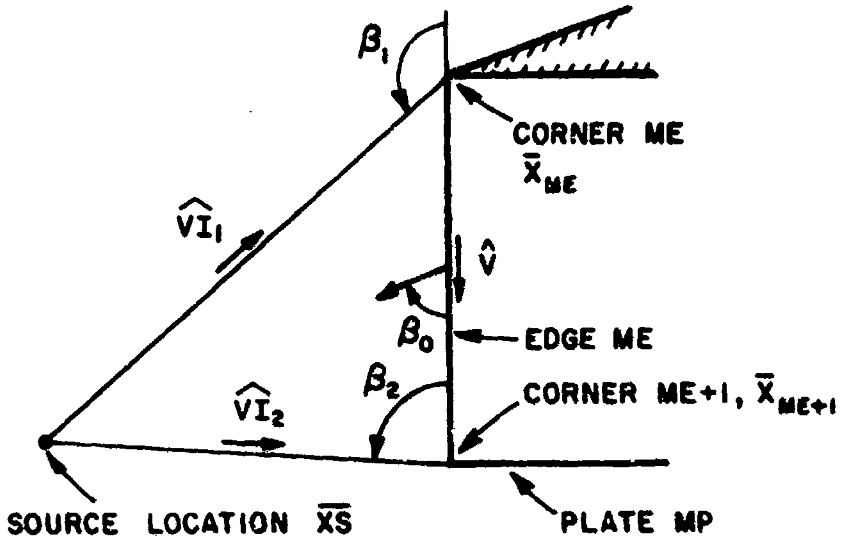


Figure 65--Geometry for determining diffraction angle range.

METHOD

The law of diffraction dictates that diffraction from a plate edge is possible when

$$\cos\beta_1 \leq \cos\beta_0 \leq \cos\beta_2,$$

where $\beta_0$ is the angle that the incident and diffracted rays make with the edge (see Figure 65). $\beta_1$ and $\beta_2$ are diffraction angle limits and are defined in terms of their cosines as:

$$BD(MP,ME,1) = \cos\beta_1 = \hat{VI}_1 \cdot \hat{V}$$

$$BD(MP,ME,2) = \cos\beta_2 = \hat{VI}_2 \cdot \hat{V},$$

where

$$\hat{VI}_1 = \frac{\overline{X}_{ME} - \overline{XS}}{|\overline{X}_{ME} - \overline{XS}|}$$

$$\hat{VI}_2 = \frac{\overline{X}_{ME+1} - \overline{XS}}{|\overline{X}_{ME+1} - \overline{XS}|} \; .$$

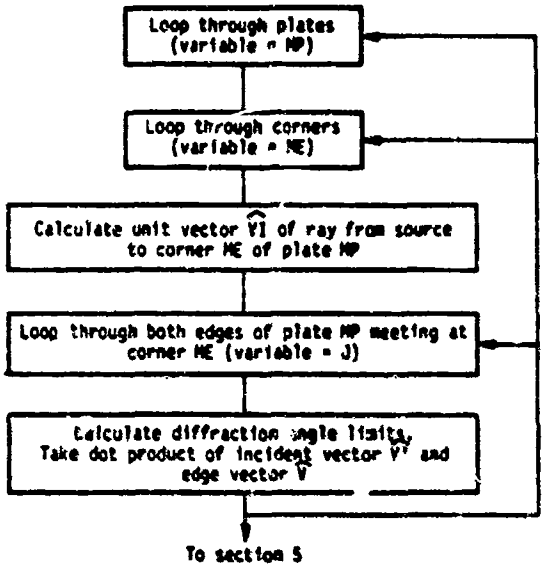
The vectors mentioned above relate to the code as follows:

$$\overline{X}_{ME} = \hat{x}\ X(MP,ME,1) + \hat{y}\ X(MP,ME,2) + \hat{z}\ X(MP,ME,3)$$

$$\overline{XS} = \hat{x}\ XS(1) + \hat{y}\ XS(2) + \hat{z}\ XS(3)$$

$$\hat{V} = \hat{x}\ V(MP,ME,1) + \hat{y}\ V(MP,ME,2) + \hat{z}\ V(MP,ME,3).$$

## FLOW DIAGRAM


Loop through plates (variable = NP)
Loop through corners (variable = NE)
Calculate unit vector $\widehat{VI}$ of ray from source to corner NE of plate NP
Loop through both edges of plate NP meeting at corner NE (variable = J)
Calculate diffraction angle limits, Take dot product of incident vector $\hat{V}'$ and edge vector $\hat{V}$
To section 5

## CODE LISTING

```
C!!!  SECTION 4  *** ****************************
C!!!  DETERMINATION OF PERMISSABLE RANGE FOR DIFFRACTION ANGLE
C!!!  LOOP THRU PLATES
      DO 42 MP=1,MPXR
      MEX=MEP(MP)
C!!!  LOOP THRU CORNERS
      DO 41 ME=1,MEX
      VIM=0.
C!!!  CALCULATE VECTOR VI FROM SOURCE TO CORNER ME OF PLATE MP
      DO 40 N=1,3
      VI(N)=X(MP,ME,N)-XS(N)
 40   VIM=VIM+VI(N)*VI(N)
      VIM=SQRT(VIM)
C!!!  LOOP THRU BOTH EDGES MEETING AT CORNER ME
      DO 41 J=1,2
      MJ=ME+1-J
      IF(MJ.EQ.0) MJ=MEX
      BD(MP,MJ,J)=0.
C!!!  CALCULATE BD, THE DOT PRODUCT OF INCIDENT RAY
C!!!  VECTOR VI AND EDGE VECTOR V
      DO 41 N=1,3
 41   BD(MP,MJ,J)=BD(MP,MJ,J)+V(MP,MJ,N)*VI(N)/VIM
 42   CONTINUE
      IF(.NOT.LDEBUG)GO TO 995
      DO 996 MP=1,MPX
      MEX=MEP(MP)
      DO 996 ME=1,MEX
 996  WRITE(6,*)(BD(MP,ME,J),J=1,2)
 995  CONTINUE
```

## GEOM SECTION 5

### PURPOSE

To calculate wedge angles for plates with common edges.

### PERTINENT GEOMETRY

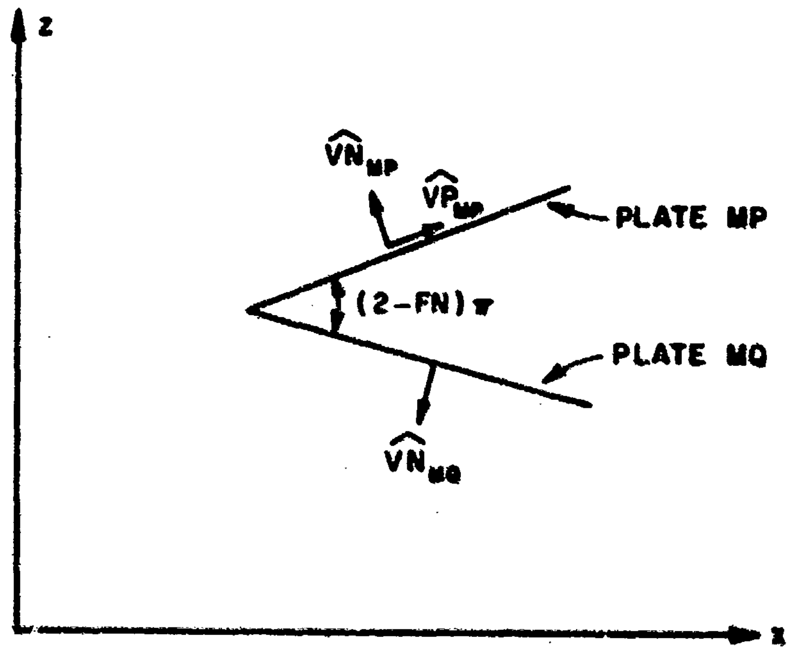


Figure 66--Geometry used to determine wedge angles of plates with common edges.

### METHOD

The wedge angle is specified using the wedge angle number FN, such that the wedge angle is given by

$$(2 - FN)\pi$$

as shown in Figure 66. The wedge angle number is determined as follows:

$$FN = \frac{1}{\pi} \tan^{-1}\left(\frac{TOP}{BOT}\right) .$$

where

$$\mathrm{TOP} = \widehat{VN}_{MQ} \cdot \widehat{VP}_{MP}$$

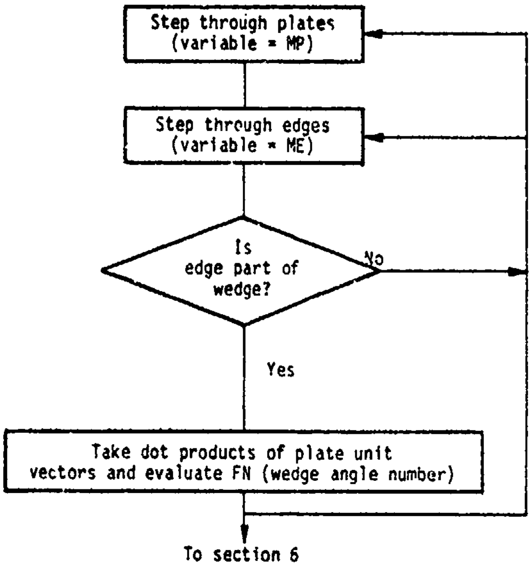
$$\mathrm{BOT} = -\widehat{VN}_{MP} \cdot \widehat{VN}_{MQ}$$

$$\widehat{VN}_{MP} = \hat{x}\ VN(MP,1) + \hat{y}\ VN(MP,2) + \hat{z}\ VN(MP,2)$$

$$\widehat{VP}_{MP} = \hat{x}\ VP(MP,ME,1) + \hat{y}\ VP(MP,ME,2) + \hat{z}\ VP(MP,ME,3)$$

$$\widehat{VN}_{MQ} = \hat{x}{=}VN(MQ,1) + \hat{y}{=}VN(MQ,2) + \hat{z}\ VN(MQ,3).$$

FLOW DIAGRAM



```
C!!!  SECTION 5  *****************************
C!!!  DETERMINATION OF WEDGE ANGLES FOR PLATES WITH COMMON EDGES
C!!!  STEP THROUGH PLATES
      DO 35 MP=1,MPX
      MEX=MEP(MP)
C!!!  STEP THROUGH EDGES
      DO 35 ME=1,MEX
C!!!  IS EDGE ME PART OF A WEDGE?
      IF(FNP(MP,ME).GT.-5.) GO TO 35
      NFN=FNP(MP,ME)-.5
      NFN=IABS(NFN)
      MQ=NFN/100
      MF=NFN-MQ*100
      IF(FNP(MQ,MF).GT.0.) GO TO 35
C!!!  TAKE DOT PRODUCTS OF PLATE UNIT VECTORS AND EVALUATE
C!!!  FN (WEDGE ANGLE NUMBER)
      BOT=-(VN(MP,1)*VN(MQ,1)+VN(MP,2)*VN(MQ,2)+VN(MP,3)*VN(MQ,3))
      TOP=VP(MP,ME,1)*VN(MQ,1)+VP(MP,ME,2)*VN(MQ,2)+
     2VP(MP,ME,3)*VN(MQ,3)
      FANG=BTAN2(TOP,BOT)
      ANN=0.
      ANP=0.
      DO 34 N=1,3
      XSX=XS(N)-X(MP,ME,N)
      ANN=ANN+XSX*VN(MP,N)
   34 ANP=ANP+XSX*VP(MP,ME,N)
      PHWAR=BTAN2(ANN,ANP)
      IF(PHWAR.LT.0.) PHWAR=TPI+PHWAR
      FN=FANG/PI
      IF(PHWAR.GT.FN*PI) FN=2.-ABS(FANG)/PI
      FNP(MP,ME)=FN
   35 CONTINUE
      IF(.NOT.LDEBUG)GO TO 997
      DO 998 MP=1,MPX
      MEX=MEP(MP)
      DO 998 ME=1,MEX
  998 WRITE(6,*)FNP(MP,ME)
  997 CONTINUE
```

## GEOM SECTION 6

PURPOSE

To determine plates which are totally shadowed from the source.

PFRTINENT GEOMETRY

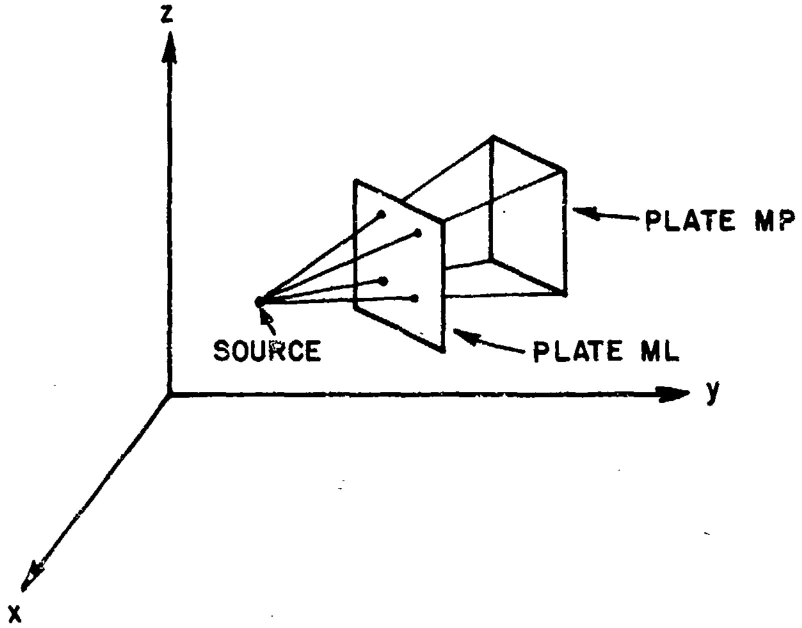


Figure 67a--Configuration where plate ML totally shadows plate MP from source.

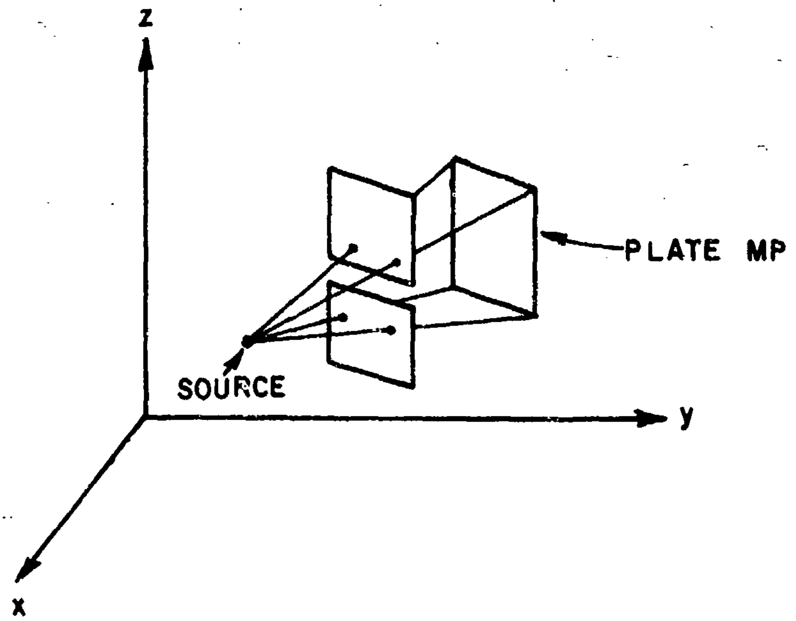
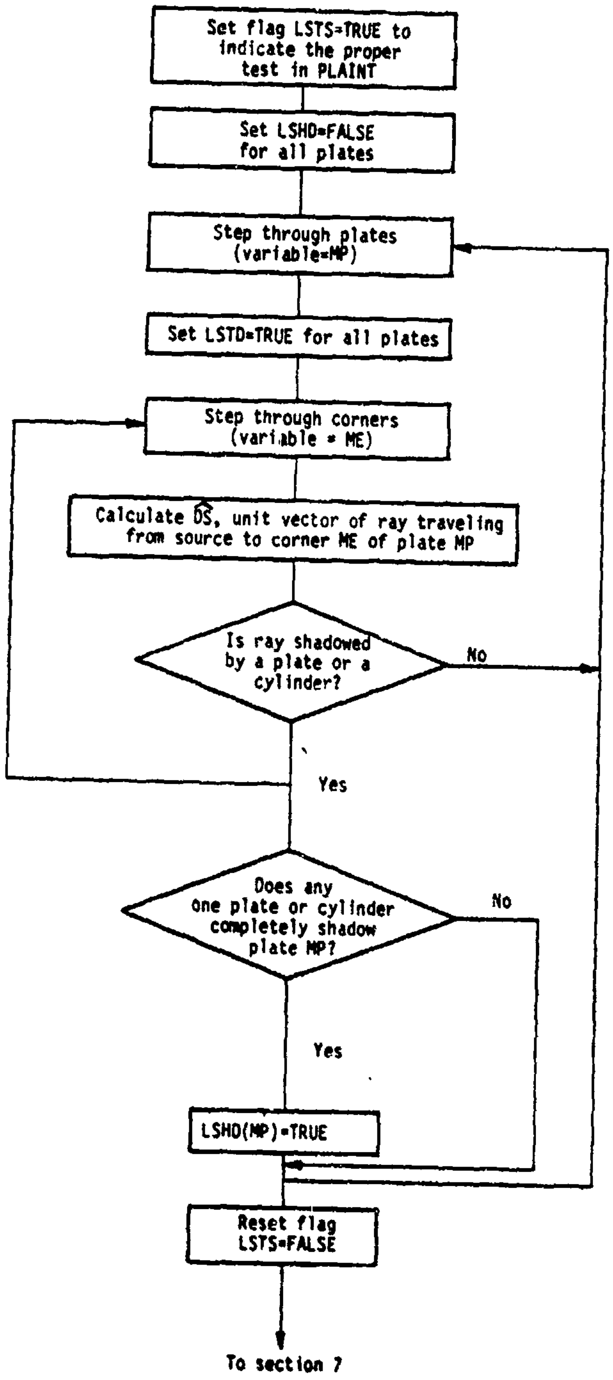


Figure 67b--Configuration where plate MP is not totally shadowed from the source

METHOD

If plate ML totally shadows plate MP from the source, then every ray drawn from the source to a corner of plate MP will intersect plate ML. The routine computes vectors from the source to each corner of plate MP and uses a shadow testing algorithm to check if any plate shadows all of the rays (see Figures 67a and 67b). If so, it is assumed that plate MP is totally shadowed from the source.

## FLOW DIAGRAM


Set flag LSTS=TRUE to indicate the proper test in PLAINT
Set LSHD=FALSE for all plates
Step through plates (variable=MP)
Set LSTD=TRUE for all plates
Step through corners (variable = ME)
Calculate $\hat{DS}$, unit vector of ray traveling from source to corner ME of plate MP
Is ray shadowed by a plate or a cylinder?
No
Yes
Does any one plate or cylinder completely shadow plate MP?
No
Yes
LSHD(MP)=TRUE
Reset flag LSTS=FALSE
To section 7

```
C!!!  SECTION 6  ******************************
C!!!  DETERMINATION OF PLATES THAT ARE TOTALLY
C!!!  SHADOWED FROM THE SOURCE
      LSTS=.TRUE.
      DO 72 MP=1,MPXR
72    LSHD(MP)=.FALSE.
C!!!  STEP THRU PLATES
      DO 79 MP=1,MPX
      MEX=MEP(MP)
C!!!  SET LSTD=TRUE FOR ALL PLATES
C!!!  SET LCTD=TRUE FOR THE CYLINDER
      DO 73 ML=1,MPX
73    LSTD(ML)=.TRUE.
      LCTD=.TRUE.
C!!!  STEP THRU CORNERS
      DO 77 ME=1,MEX
      DSM=0.
C!!!  CALCULATE DS, UNIT VECTOR OF RAY TRAVELING FROM SOURCE TO
C!!!  CORNER ME OF PLATE MP
      DO 74 N=1,3
      DS(N)=X(MP,ME,N)-XS(N)
74    DSM=DSM+DS(N)*DS(N)
      DSM=SQRT(DSM)
      DO 75 N=1,3
75    DS(N)=DS(N)/DSM
C!!!  IS RAY SHADOWED BY PLATE OR CYLINDER?
      CALL PLAINT(XS,DS,DHIT,MP,LHIT)
      IF(LHIT.AND.DHIT.GT.DSM)LHIT=.FALSE.
      IF(.NOT.LCTD) GO TO 76
      PHCR=BTAN2(DS(2),DS(1))
      CALL CYLINT(XS,DS,PHCR,DHIT,LHCT,.FALSE.)
      IF(.NOT.LHCT) LCTD=.FALSE.
      IF(LHCT.AND.DHIT.GT.DSM) LCTD=.FALSE.
76    CONTINUE
      IF(.NOT.LHIT.AND..NOT.LCTD) GO TO 79
77    CONTINUE
C!!!  CHECK TO SEE IF ANY ONE PLATE ML COMPLETELY SHADOWS PLATE MP
C!!!  STEP THRU PLATES
      DO 78 ML=1,MPX
      IF(.NOT.LSTD(ML)) GO TO 78
      LSHD(MP)=.TRUE.
78    CONTINUE
      IF(LCTD) LSHD(MP)=.TRUE.
79    CONTINUE
      IF (LDEBUG) WRITE (6,*) (LSHD(MP),MP=1,MPXR)
      LSTS=.FALSE.
```

## GEOM SECTION 7

### PURPOSE

This section handles various calculations for plates which intersect each other.

### PERTINENT GEOMETRY

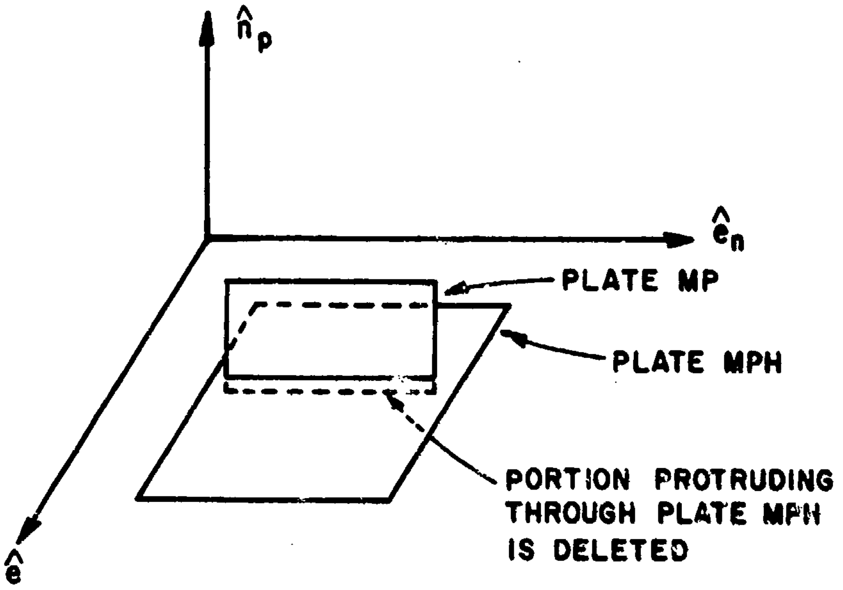
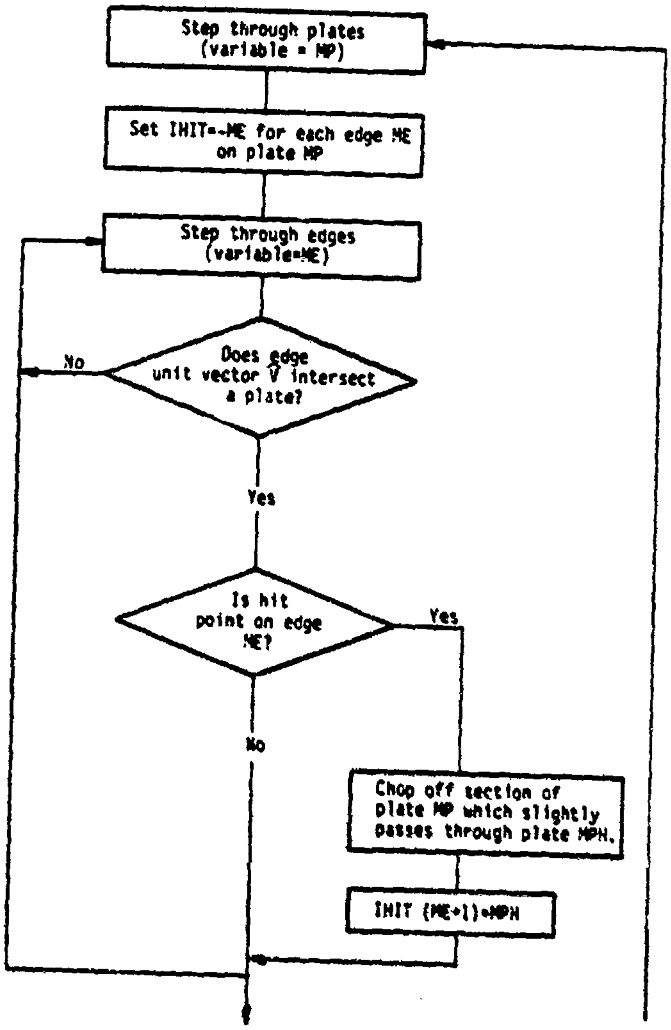


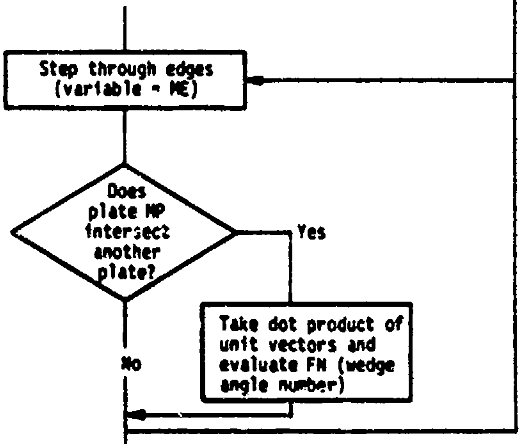
Figure 68--Illustration of a plate which intersects another.

## FLOW DIAGRAM

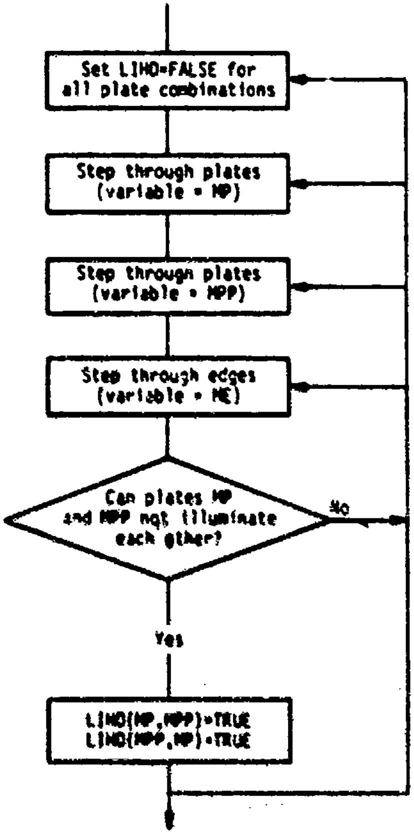
1. Determine plates that intersect


Step through plates (variable = MP)
Set INIT=-ME for each edge ME on plate MP
Step through edges (variable=ME)
Does edge unit vector V intersect a plate?
No
Yes
Is hit point on edge ME?
Yes
No
Chop off section of plate MP which slightly passes through plate MPH.
INIT (ME+1)=MPH

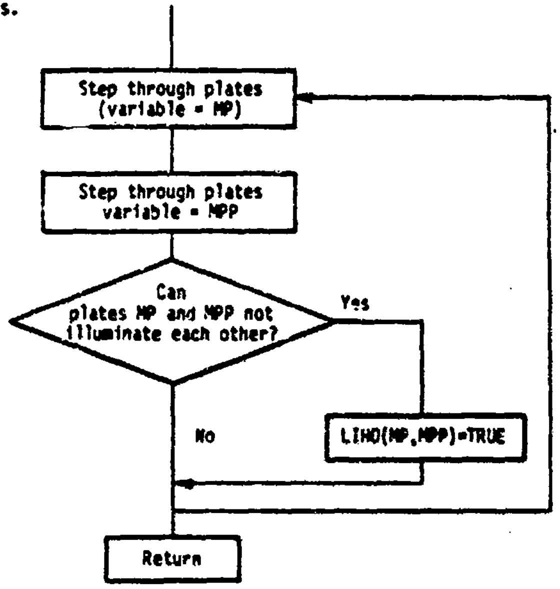
2. Determine new wedge angle number for intersecting plates


Step through edges (variable = ME)
Does plate MP intersect another plate?
Yes
Take dot product of unit vectors and evaluate FN (wedge angle number)
No

3. Find plates with common edges which cannot illuminate each other because wedge angle between plates is greater than 180°.


Set LIHO=FALSE for all plate combinations
Step through plates (variable = MP)
Step through plates (variable = MPP)
Step through edges (variable = ME)
Can plates MP and MPP not illuminate each other?
No
Yes
LIHO(MP,MPP)=TRUE
LIHO(MPP,MP)=TRUE

4. Determine which plates cannot illuminate each other based on illegal image locations.


Step through plates (variable = MP)
Step through plates variable = MPP
Can plates MP and MPP not illuminate each other?
Yes
No
LIHO(MP,MPP)=TRUE
Return

## CODE LISTING

```
C!!!  SECTION 7  ******************************
C!!!  1. DETERMINE PLATES THAT INTERSECT
C!!!  STEP THRU PLATES
      DO 85 MP=1,MPX
      MEX=MEP(MP)
C!!!  SET IHIT=-ME FOR EACH EDGE ON PLATE MP
      DO 889 ME=1,MEX
  889 IHIT(ME)=-ME
C!!!  STEP THRU EDGES
      DO 80 ME=1,MEX
      DO 81 N=1,3
      XIN(N)=X(MP,ME,N)
   81 DS(N)=V(MP,ME,N)
C!!!  DOES EDGE INTERSECT ANOTHER PLATE?
      CALL PLAINT(XIN,DS,DHIT,MP,LHIT)
      IF(.NOT.LHIT)GO TO 80
      IF(DHIT.GT.VMAG(MP,ME))GO TO 80
      MC=ME+1
      IF(MC.GT.MEX)MC=1
C!!!  CHOP OFF SECTOR OF PLATE MP WHICH PASSES THRU PLATE MPH
      IF(DHIT.LT.0.1)GO TO 83
      DO 82 N=1,3
   82 X(MP,MC,N)=X(MP,ME,N)+(DHIT-2.E-5)*V(MP,ME,N)
      VMAG(MP,ME)=DHIT-2.E-5
      IHIT(MC)=MPH
      IF(LDEBUG)WRITE(6,*)MP,MC,(X(MP,MC,N),N=1,3)
      GO TO 80
   83 VMAG(MP,ME)=VMAG(MP,ME)-DHIT
      DO 84 N=1,3
   84 X(MP,ME,N)=X(MP,MC,N)-VMAG(MP,ME)*V(MP,ME,N)
      IHIT(ME)=MPH
      IF(LDEBUG)WRITE(6,*)MP,ME,(X(MP,ME,N),N=1,3)
   80 CONTINUE
C!!!  2. DETERMINE NEW WEDGE ANGLE NUMBER FOR INTERSECTING PLATES
C!!!  STEP THRU EDGES
      DO 86 ME=1,MEX
      MC=ME+1
      IF(MC.GT.MEX)MC=1
C!!!  DO PLATES INTERSECT?
      IF(IHIT(ME).NE.IHIT(MC))GO TO 86
      MR=IHIT(ME)
C!!!  TAKE DOT PRODUCTS OF PLATE UNIT VECTORS AND EVALUATE FN
      XX=VN(MR,1)*VP(MP,ME,1)+VN(MR,2)*VP(MP,ME,2)+VN(MR,3)
     2*VP(MP,ME,3)
      YY=VN(MR,1)*VN(MP,1)+VN(MR,2)*VN(MP,2)+VN(MR,3)*VN(MP,3)
      FN=-MR
      IF(XX.LE.0.)GO TO 89
      FN=0.5+ATAN2(YY,XX)/PI
      ANN=0.
      DO 87 N=1,3
   87 ANN=ANN+VN(MP,N)*(XS(N)-X(MP,ME,N))
      IF(ANN.GT.0.)GO TO 88
      FN=3.-FN
      GO TO 88
   89 WRITE(6,881)MP,MR
  881 FORMAT(' WARNING PLATES ',2I5,' INTERSECT YET GEOMETRY '
     3,'INDICATES ATTACHED PLATE IS SHADOWED!!!',//)
   88 FNP(MP,ME)=FN
   86 CONTINUE
   85 CONTINUE
      IF(.NOT.LDEBUG)GO TO 887
      DO 888 MP=1,MPX
      MEX=MEP(MP)
      DO 888 ME=1,MEX
  888 WRITE(6,*)FNP(MP,ME)
  887 CONTINUE
```

```
C!!!  3. DETERMINE PLATES WITH COMMON EDGES WHICH CANNOT ILLUMINATE
C!!!  EACH OTHER.
C!!!  SET LIHD=FALSE FOR ALL PLATE COMBINATIONS
      DO 9Ø MP=1,MPXR
      DO 9Ø MPP=1,MPXR
9Ø    LIHD(MP,MPP)=.FALSE.
C!!!  STEP THRU PLATES
      DO 91 MP=1,MPX
C!!!  STEP THRU PLATES
      DO 91 MPP=1,MPX
      MEX=MEP(MPP)
C!!!  STEP THRU EDGES
      DO 92 ME=1,MEX
C!!!  CAN PLATES MP AND MPP NOT ILLUMINATE EACH OTHER?
C!!!  IF SO, IDENTIFY
      IFN=-FNP(MPP,ME)/1ØØ.
      IF(IFN.NE.MP)GO TO 92
      MEH=-FNP(MPP,ME)-IFN*1ØØ+Ø.5
      IF(FNP(IFN,MEH).LT.1.)GO TO 92
      LIHD(MP,MPP)=.TRUE.
      LIHD(MPP,MP)=.TRUE.
92    CONTINUE
91    CONTINUE
C!!!  4. DETERMINE PLATES WHICH CANNOT ILLUMINATE EACH OTHER BASED
C!!!  ON ILLEGAL IMAGE LOCATIONS.
C!!!  STEP THRU PLATES
      DO 921 MP=1,MPX
      MEX=MEP(MP)
      SUMT=1.E3Ø
      DO 922 ME=1,MEX
      SUM=Ø.
      DO 923 N=1,3
923   SUM=SUM+(X(MP,ME,N)-XS(N))**2
      IF(SUM.GT.SUMT)GO TO 922
      SUMT=SUM
      MEE=ME
922   CONTINUE
      DO 924 MPP=1,MPX
      ANP=Ø.
      ANI=Ø.
      DO 925 N=1,3
      ANP=ANP+(X(MP,MEE,N)-X(MPP,1,N))*VN(MPP,N)
925   ANI=ANI+(XI(MP,MPP,N)-X(MPP,1,N))*VN(MPP,N)
C!!!  CAN PLATES MP AND MPP NOT ILLUMINATE EACH OTHER?
      IF(ANP*ANI.LT.Ø.)GO TO 924
      LIHD(MP,MPP)=.TRUE.
924   CONTINUE
921   CONTINUE
      IF(.NOT.LDEBUG)GO TO 93
      DO 94 MP=1,MPXR
      DO 94 MPP=1,MPXR
94    WRITE(6,*)MP,MPP,LIHD(MP,MPP)
93    CONTINUE
      RETURN
      END
```

## SYMBOL DICTIONARY

AN DOT PRODUCT OF PLATE UNIT NORMAL AND VECTOR FROM SOURCE TO THE PLATE (CALCULATED IN IMAGE)

ANI DOT PRODUCT OF VECTOR FROM CORNER 1 OF PLATE MPP TO THE DOUBLE REFLECTION IMAGE LOCATION XI(MP,MPP) AND THE UNIT NORMAL OF PLATE MPP

ANN DOT PRODUCT OF XSX AND UNIT NORMAL OF PLATE MP

ANP DOT PRODUCT OF VECTOR FROM CORNER 1 OF PLATE MPP TO CORNER MEE OF PLATE MP AND UNIT NORMAL OF PLATE MPP
ALSO DOT PRODUCT OF XSX AND BINORMAL OF EDGE ME OF PLATE MP

BD COSINES OF ANGLES DEFINING BOUNDS ON DIFFRACTION ANGLE

DOT NEGATIVE DOT PRODUCT OF UNIT NORMALS OF PLATES MP AND MQ

DHIT DISTANCE FROM SOURCE TO NEAREST HIT (FROM PLAINT)

DOT DOT PRODUCT OF PLATE UNIT NORMAL AND EDGE UNIT NORMAL

DS UNIT VECTOR OF RAY FROM SOURCE TO CORNER ME OF PLATE MP (SECTION 6)
ALSO UNIT VECTOR OF RAY FROM IMAGE TO SOURCE (SECTION 3)
ALSO UNIT VECTOR OF EDGE ME (SECTION 7)

DSM NORMALIZATION CONSTANT FOR DS

ENORM DOT PRODUCT OF VN (THE UNIT NORMAL OF PLATE MP) AND THE Z AXIS OF THE SOURCE COORDINATE SYSTEM

FACTOR MAGNITUDE ADJUSTMENT FOR SOURCES MOUNTED ON THE SURFACE OF PLATES

FANG WEDGE ANGLE

FN WEDGE ANGLE NUMBER

FNP WEDGE ANGLE NUMBER (ALSO USED IN DEFINING COMMON EDGES)

IFN INDEX VARIABLE

INTT STORES PLATE NUMBERS FOR PLATES INTERSECTED BY AN EDGE

J DO LOOP VARIABLE

LCHT SET TRUE IF RAY HITS CYLINDER (RETURNED FROM CYLINT)

LCTD SET TRUE IF CYLINDER SHADOWS PLATE FROM SOURCE

LHIT SET TRUE IF RAY INTERSECTS A PLATE (FROM PLAINT)

LIMU SET TRUE IF PLATES MP AND MPP CANNOT ILLUMINATE EACH OTHER

LSHD SET TRUE IF PLATE MP IS TOTALLY SHADOWED FROM THE SOURCE

LSTD SET TRUE IF PLATE ML TOTALLY SHADOWS PLATE MP FROM THE SOURCE

LSTS SET TRUE IF TOTAL SHADOWING ALGORITHM IS BEING USED

MA INDEX VARIABLE

ME DO LOOP VARIABLE, ALSO INDEX VARIABLE

MEC WORKING VARIABLE

MEE INDEX VARIABLE

MEH INDEX VARIABLE

MEN WORKING VARIABLE

MES COMPUTATIONAL VARIABLE

MEX NUMBER OF EDGES ON A GIVEN PLATE

MP DO LOOP VARIABLE, ALSO INDEX VARIABLE

MPC WORKING VARIABLE

MPN WORKING VARIABLE

MPS COMPUTATIONAL VARIABLE

MPX NUMBER OF EDGES ON PLATE MQ

ML DO LOOP VARIABLE

MQL INDEX VARIABLE

NP DO LOOP VARIABLE (STEP THRU PLATES)
ALSO PLATE INDEX VARIABLE

| | |
|---|---|
| MPP | DO LOOP VARIABLE (ALSO PLATE INDEX VARIABLE) |
| MO | DO LOOP VARIABLE (STEP THRU PLATES).<br>ALSO INDEX VARIABLE |
| MR | INDEX VARIABLE |
| MV | INDEX VARIABLE |
| N | DO LOOP VARIABLE |
| NFL | WORKING VARIABLE |
| NI | DO LOOP VARIABLE |
| NJ | DO LOOP VARIABLE |
| PHCR | PHI COMPONENT OF VECTOR FROM SOURCE TO PLATE CORNER IN RCS |
| PHRAN | ANGLE WHICH DETERMINES WHICH SIDE OF THE INTERSECTING PLATES IS ILLUMINATED |
| SUR | LENGTH OF VECTOR FROM SOURCE TO EDGE ME OF PLATE MP |
| SURT | LENGTH OF VECTOR FROM SOURCE TO CLOSEST EDGE OF PLATE MP |
| TOP | DOT PRODUCT OF BINORMAL OF COMMON EDGE OF PLATE MP AND NORMAL OF PLATE MO |
| V | MATRIX OF X,Y,Z COMPONENTS DEFINING EDGE UNIT VECTORS IN RCS |
| VAX | X,Y,Z COMPONENTS DEFINING SINGLE REFLECTION IMAGE SOURCE COORDINATE SYSTEM AXES IN RCS COMPONENTS |
| VI | X,Y,Z COMPONENTS OF UNIT VECTOR OF RAY FROM SOURCE TO CORNER ME OF PLATE MP |
| VIM | NORMALIZATION CONSTANT OF VI |
| VM | DISTANCE BETWEEN TWO NEIGHBORING CORNERS ON A PLATE SQUARED |
| VMAG | DISTANCES BETWEEN NEIGHBORING CORNERS ON PLATES |
| VN | X,Y,Z COMPONENTS DEFINING PLATE UNIT NORMAL DIRECTIONS IN RCS COMPONENTS |
| VNA | PLATE UNIT NORMAL NORMALIZATION CONSTANT |
| VP | MATRIX CONTAINING EDGE UNIT BINORMAL DIRECTIONS IN REFERENCE COORDINATE SYSTEM |
| VXI | X,Y,Z COMPONENTS DEFINING UNIT VECTORS OF SOURCE IMAGE COORDINATE SYSTEM AXES IN RCS |
| VXS | X,Y,Z COMPONENTS DEFINING UNIT VECTORS OF SOURCE COORDINATE SYSTEM AXES IN RCS |
| XI | ARRAY OF COMPONENTS DEFINING SINGLE AND DOUBLE REFLECTION SOURCE IMAGE LOCATIONS IN RCS |
| XII | X,Y,Z COMPONENTS OF SINGLE REFLECTION IMAGE LOCATION CALCULATED IN SUBROUTINE IMAGE |
| XIN | X,Y,Z COMPONENTS OF LOCATION OF CORNER ME OF PLATE MP<br>ALSO, SINGLE REFLECTION IMAGE LOCATION<br>ALSO, XSII |
| XM | DISTANCE BETWEEN CORNER MF ON PLATE MO AND CORNER ME ON PLATE MP |
| XSI | X,Y,Z COMPONENTS OF SOURCE LOCATION MOVED A SMALL AMOUNT IN THE DIRECTION OF THE PLATE NORMAL (FOR SOURCE ON PLATE PLANE) |
| XSII | X,Y,Z COMPONENTS OF SOURCE IMAGE LOCATION CALCULATED IN SUBROUTINE IMAGE FOR SOURCE LOCATED AT XSI |
| XSX | X,Y,Z COMPONENTS OF VECTOR FROM CORNER ME OF PLATE MP TO THE SOURCE |
| XX | DOT PRODUCT OF BINORMAL OF EDGE ME OF PLATE MP AND NORMAL OF PLATE MO |
| YY | DOT PRODUCT OF NORMALS OF PLATES MP AND MO |

GEOMC

PURPOSE

To calculate geometry associated with fixed cylinder structures (end cap normals, etc.).

PERTINENT GEOMETRY

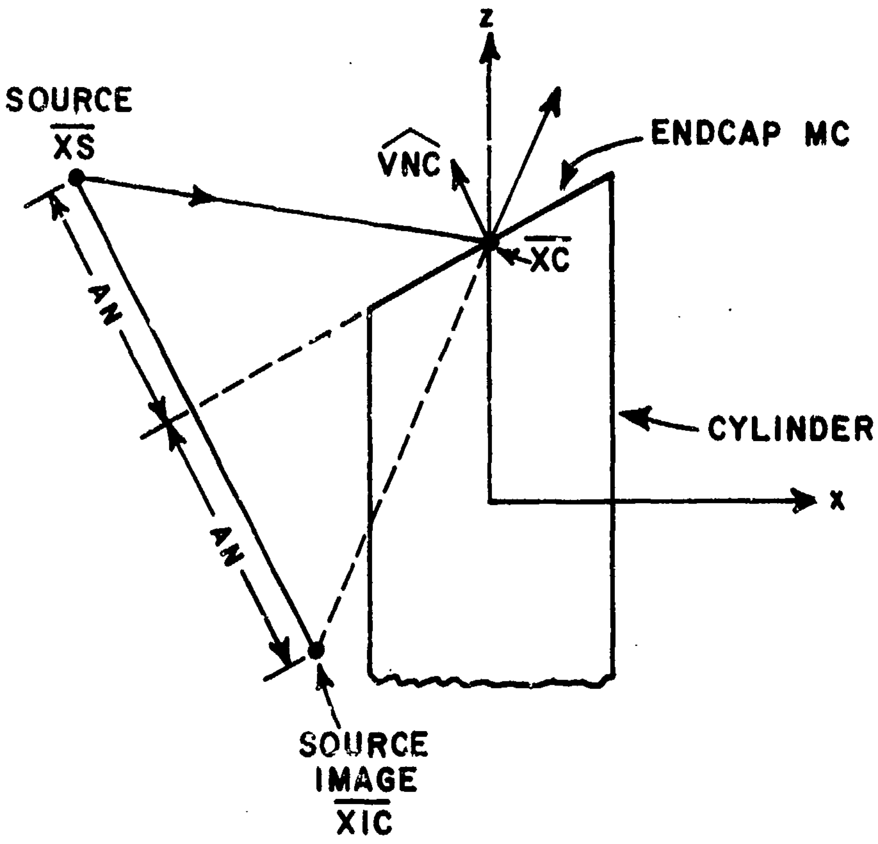


Figure 69-- Geometry for determining source image location for reflection from cylinder end cap.

$\widehat{VNC} = \hat{x}\ VNC(1) + \hat{y}\ VNC(2) + \hat{z}\ VNC(3)$

$\overline{XS} = \hat{x}\ XS(1) + \hat{y}\ XS(2) + \hat{z}\ XS(3)$

$\overline{XIC} = \hat{x}\ XIC(MC,1) + \hat{y}\ XIC(MC,2) + \hat{z}\ XIC(MC,3)$

$\overline{XC} = \hat{z}\ ZC(MC)$

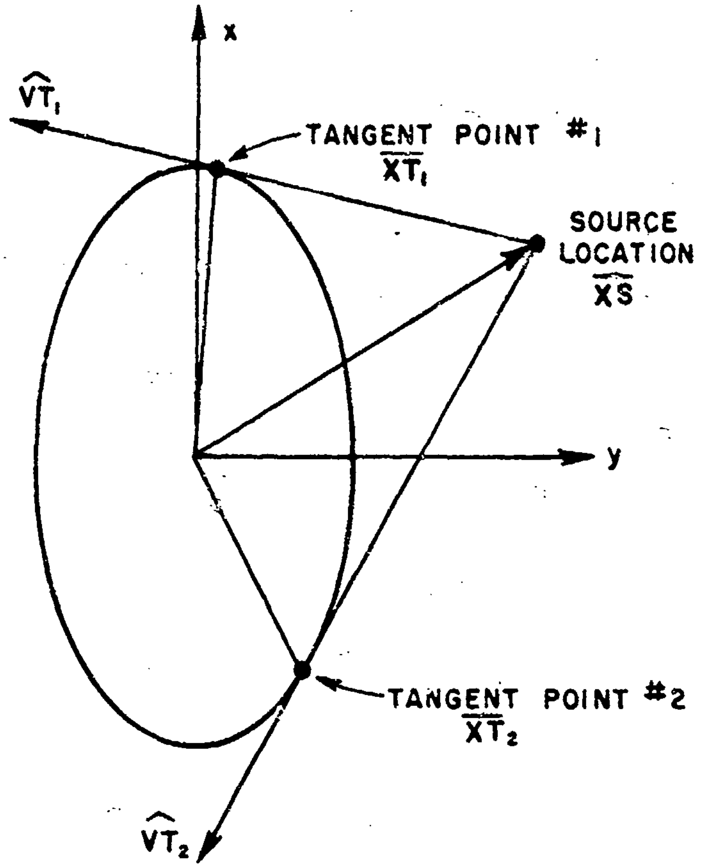


Figure 70-- Illustration of vectors from the source tangent to the elliptic cylinder.

$$\widehat{VT}_1 = \hat{x}\ BTS(1) + \hat{y}\ BTS(2)$$

$$\widehat{VT}_2 = \hat{x}\ BTS(3) + \hat{y}\ BTS(4)$$

$$\overline{XT}_1 = \hat{x}\ A\cos(VTS(1)) + \hat{y}\ B\sin(VTS\ (1))$$

$$\overline{XT}_2 = \hat{x}\ A\cos(VTS(2)) + \hat{y}\ B\sin(VTS(2))$$

$$\overline{XS} = \hat{x}\ XS(1) + \hat{y}\ XS(2) + \hat{z}\ XS(3)$$

METHOD

The image source location is given by:

$\overline{XIC} = \overline{XS} - 2\ AN\ \widehat{VNC}$,

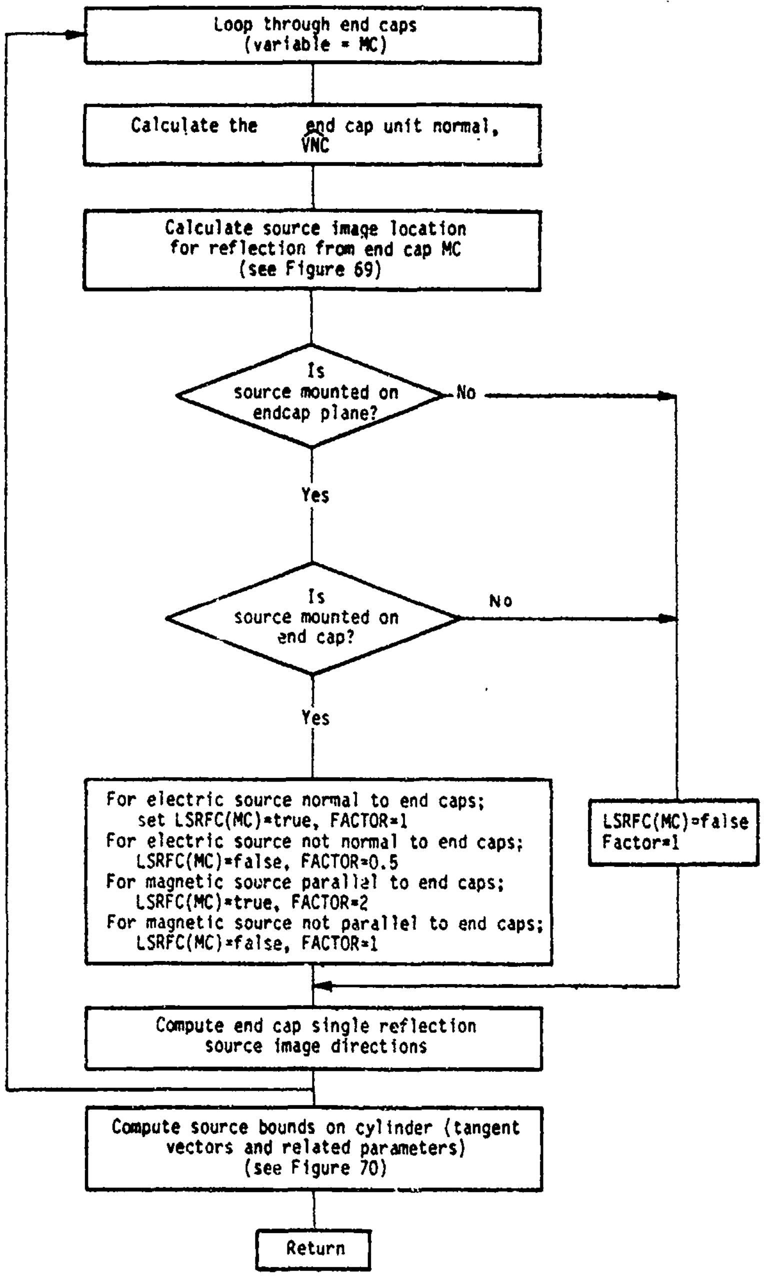
where

$$AN = (\overline{XS} - \overline{XC}) \cdot \widehat{VNC} .$$

This is illustrated in Figure 69.

The tangent vectors from the source to the cylinder, as illustrated in Figure 70, are found in subroutine TANG.

FLOW DIAGRAM


Loop through end caps (variable = MC)
Calculate the end cap unit normal, VNC
Calculate source image location for reflection from end cap MC (see Figure 69)
Is source mounted on endcap plane?
No
Yes
Is source mounted on end cap?
No
Yes
For electric source normal to end caps; set LSRFC(MC)=true, FACTOR=1
For electric source not normal to end caps; LSRFC(MC)=false, FACTOR=0.5
For magnetic source parallel to end caps; LSRFC(MC)=true, FACTOR=2
For magnetic source not parallel to end caps; LSRFC(MC)=false, FACTOR=1
LSRFC(MC)=false Factor=1
Compute end cap single reflection source image directions
Compute source bounds on cylinder (tangent vectors and related parameters) (see Figure 70)
Return

## SYMBOL DICTIONARY

| | |
|---|---|
| AN | DOT PRODUCT OF END CAP NORMAL AND RAY FROM END CAP TO SOURCE |
| DS | UNIT VECTOR OF RAY FROM SOURCE IMAGE TO SOURCE |
| DSM | DISTANCE FROM SOURCE IMAGE TO SOURCE |
| ENORM | DOT PRODUCT OF END CAP NORMAL AND Z AXIS OF SOURCE COORDINATE SYSTEM |
| LHIT | SET TRUE IF RAY HITS END CAP (FROM SUB. CAPINT) |
| LSRFC | SET TRUE IF SOURCE IS MOUNTED ON END CAP MC |
| MC | END CAP INDEX VARIABLE |
| N | DO LOOP VARIABLE |
| NC | SIGN CHANGE VARIABLE |
| NI | DO LOOP VARIABLE |
| NJ | DO LOOP VARIABLE |
| VNC | X,Y, AND Z COMPONENTS OF THE END CAP UNIT NORMAL IN REF COORD SYS |
| VXIC | X,Y,Z COMPONENTS OF UNIT VECTORS DEFINING AXES OF END CAP SOURCE IMAGE COORDINATE SYSTEM |
| XIN | SOURCE IMAGE LOCATION IN END CAP MC |

## CODE LISTING

```
C-----------------------------------------------------------------------
      SUBROUTINE GEOMC
C!!!
C!!!  THIS ROUTINE COMPUTES ALL THE GEOMETRY ASSOCIATED
C!!!  WITH FIXED CYLINDER STRUCTURES, END CAP NORMALS, ETC.
C!!!
      DIMENSION XIN(3),DS(3),VNC(3),VAX(3,3)
      LOGICAL LPLA,LCYL,LSRFC,LHIT,LDEBUG,LTEST
      COMMON/PIS/PI,TPI,DPR,RPD
      COMMON/GEOMEL/A,B,ZC(2),SNC(2),CNC(2),CTC(2)
      COMMON/SORINF/XS(3),VXS(3,3)
      COMMON/IMCINF/XIC(2,3),VXIC(3,3,2)
      COMMON/FARP/IM,H,HAW
      COMMON/SOURSF/FACTOR
      COMMON/ENDSCL/DTS,VTS(2),BTS(4)
      COMMON/SRFACC/LSRFC(2)
      COMMON/LPLCY/LPLA,LCYL
      COMMON/TEST/LDEBUG,LTEST
      IF(LDEBUG) WRITE(6,900)
  900 FORMAT(/,' DEBUGGING GEOMC SUBROUTINE')
C!!!  DETERMINATION OF DISK IMAGES
      IF(.NOT.LPLA) FACTOR=1.
C!!!  LOOP THRU END CAPS
      DO 515 MC=1,2
      LSRFC(MC)=.FALSE.
      NC=MC
      IF(MC.EQ.2) NC=-1
C!!!  CALCULATE END CAP UNIT NORMAL
      VNC(1)=-NC*CNC(MC)
      VNC(2)=0.
      VNC(3)=NC*SNC(MC)
C!!!  CALCULATE SOURCE IMAGE LOCATION FOR REFLECTION FROM
C!!!  END CAP MC
      AN=XS(1)*VNC(1)+XS(2)*VNC(2)+(XS(3)-ZC(MC))*VNC(3)
      DO 510 N=1,3
  510 XIC(MC,N)=XS(N)-2.*AN*VNC(N)
C!!!  IS SOURCE MOUNTED ON END CAP PLANE?
      IF(ABS(AN).GT.1.E-5) GO TO 520
      DO 526 N=1,3
  526 XS(N)=XS(N)+1.E-5*VNC(N)
      AN=XS(1)*VNC(1)+XS(2)*VNC(2)+(XS(3)-ZC(MC))*VNC(3)
      DO 527 N=1,3
  527 XIC(MC,N)=XS(N)-2.*AN*VNC(N)
C!!!  IS ANTENNA MOUNTED ON END CAP, IF SO IDENTIFY
      DSM=0.
      DO 523 N=1,3
      DS(N)=XS(N)-XIC(MC,N)
  523 DSM=DSM+DS(N)*DS(N)
      DSM=SQRT(DSM)
      DO 524 N=1,3
      DS(N)=DS(N)/DSM
  524 XIN(N)=XIC(MC,N)
      CALL CAPINT(XIN,DS,DHIT,MC,LHIT)
      IF(.NOT.LHIT) GO TO 520
      ENORM=VNC(1)*VXS(3,1)+VNC(2)*VXS(3,2)+VNC(3)*VXS(3,3)
      IF(IM.NE.0) GO TO 521
      IF(1.-ABS(ENORM).GT.1.E-3) GO TO 522
      LSRFC(MC)=.TRUE.
      GO TO 520
  522 FACTOR=.5
      GO TO 520
  521 IF(ABS(ENORM).GT.1.E-3) GO TO 520
      LSRFC(MC)=.TRUE.
      FACTOR=2.
  520 CONTINUE
```

```
C!!!  COMPUTE END CAP IMAGE SOURCE AXES DIRECTIONS
      CALL IMCDIR(VAX,VXS,VNC)
      DO 530 NJ=1,3
      DO 530 NI=1,3
530   VXIC(NI,NJ,MC)=VAX(NI,NJ)
515   CONTINUE
      IF(.NOT.LDEBUG) GO TO 910
      DO 911 MC=1,2
      WRITE(6,*) MC,LSRFC(MC)
      WRITE(6,*) (XIC(MC,N),N=1,3)
      DO 912 NI=1,3
912   WRITE(6,*) NI,(VXIC(NI,NJ,MC),NJ=1,3)
911   CONTINUE
910   CONTINUE
C!!!  DETERMINATION OF SOURCE BOUNDS ON CYLINDER
      CALL TANG(DTS,VTS,BTS,XS)
      IF(.NOT.LDEBUG) GO TO 915
      WRITE(6,*) DTS
      WRITE(6,*)VTS(1),VTS(2)
      WRITE(6,*)(BTS(J),J=1,4)
915   CONTINUE
      RETURN
      END
```

## GEOMPC

PURPOSE

To compute variables pertaining to plate-cylinder interactions which are constant for a given set of plates and cylinder and a given source.

PERTINENT GEOMETRY

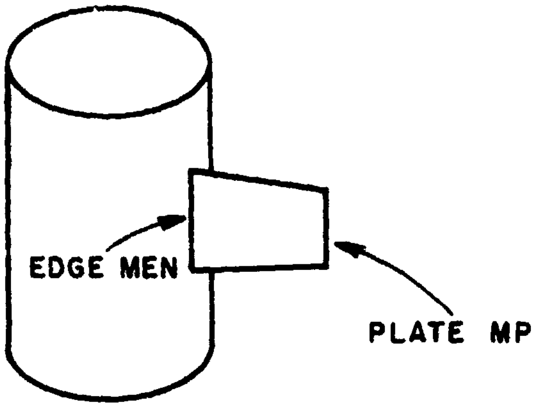


Figure 71-- Illustration of plate attached to cylinder as detailed in section 1.

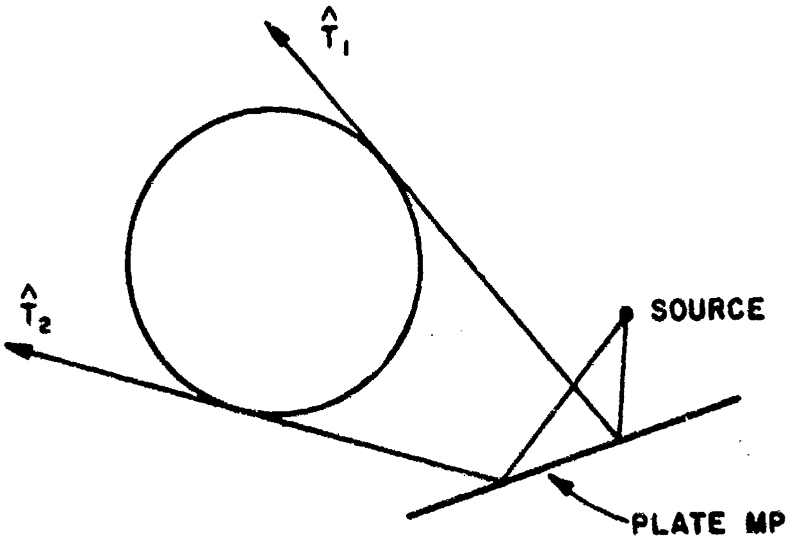


Figure 72-- Illustration of source rays reflected by plate MP tangent to the cylinder as detailed in section 2.

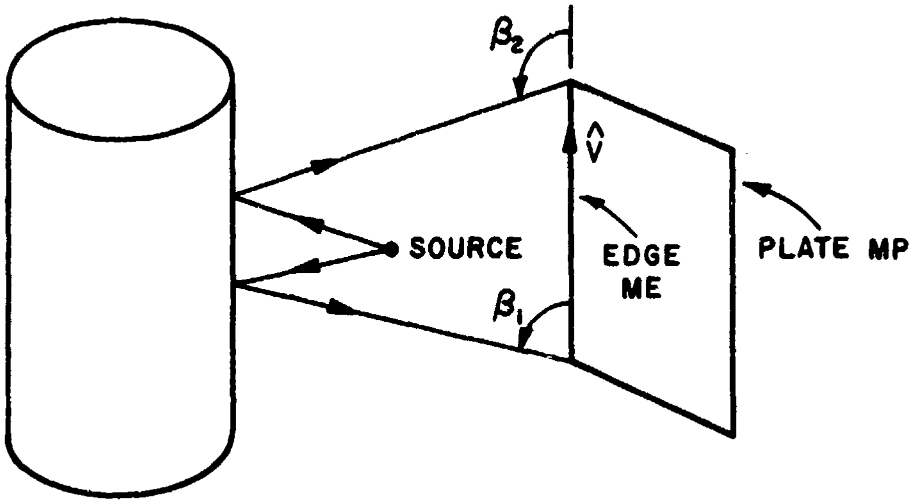


Figure 73-- Illustration of bounds for cylinder reflected, plate diffracted region detailed in section 3.

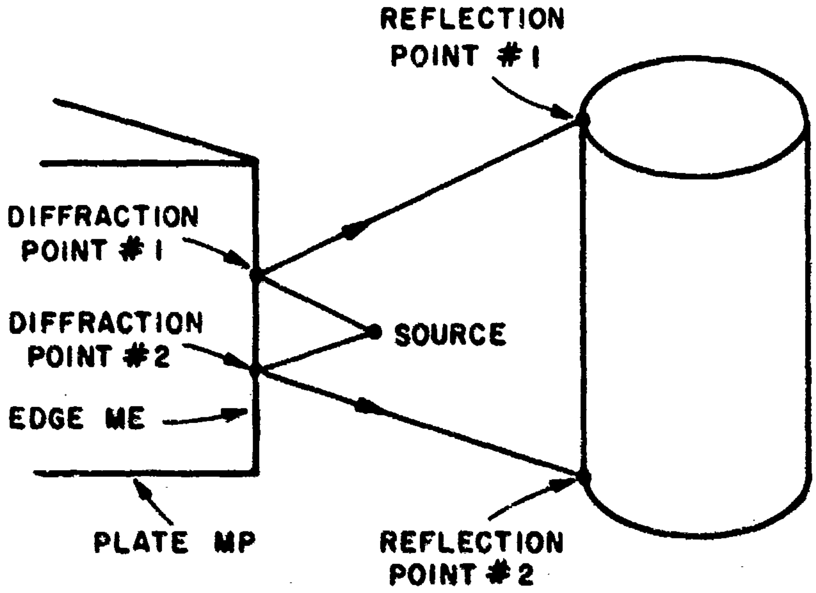


Figure 74-- Illustration of starting point path for plate diffracted, cylinder reflected ray tracing algorithm as detailed in section 4.

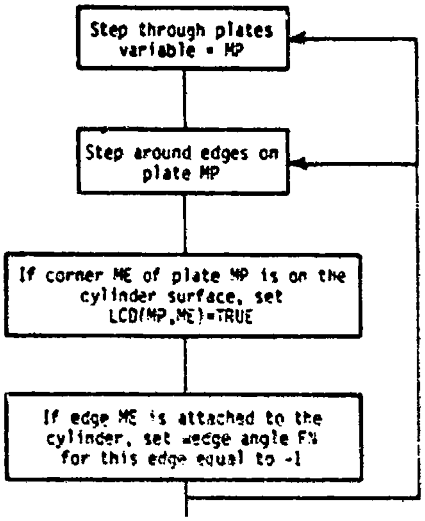
METHOD

The bounds for cylinder reflected, plate diffracted fields are illustrated in Figure 73. Details of the method used to find these parameters are given on pages 149-154 of Reference 1. Also see the write-up for subroutine RFDFPT. The starting point paths for plate diffracted, cylinder reflected fields are illustrated in Figure 74. Details of the method used to find these parameters are given on pages 161-163 of Reference 1. Also see the write-up for subroutine DFRFPT.

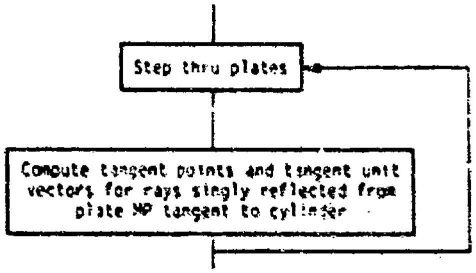
# FLOW DIAGRAM

FLOW DIAGRAM

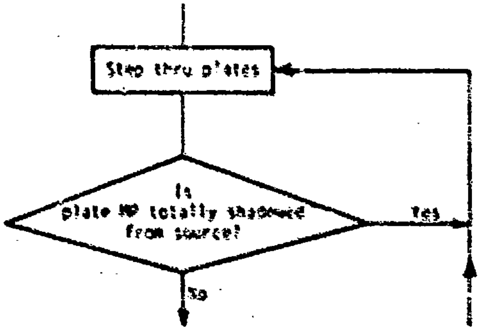
1. Determine corners and edges which are attached to the cylinder.

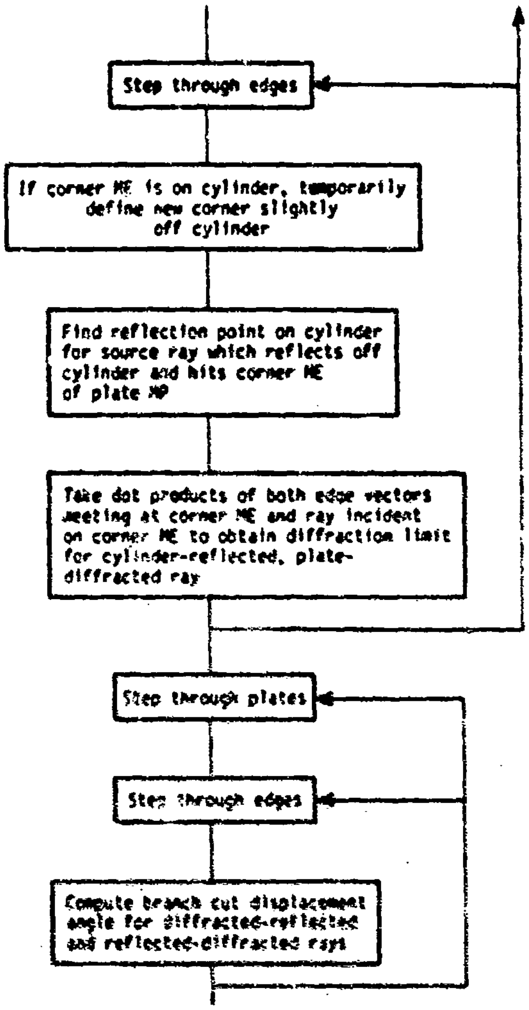

Step through plates variable = MP
Step around edges on plate MP
If corner ME of plate MP is on the cylinder surface, set LCD(MP,ME)=TRUE
If edge ME is attached to the cylinder, set wedge angle FN for this edge equal to -1

2. Determination of image bounds on cylinder

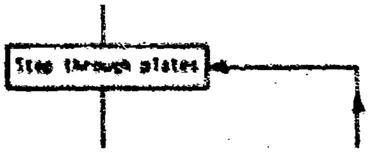

Step thru plates
Compute tangent points and tangent unit vectors for rays singly reflected from plate MP tangent to cylinder

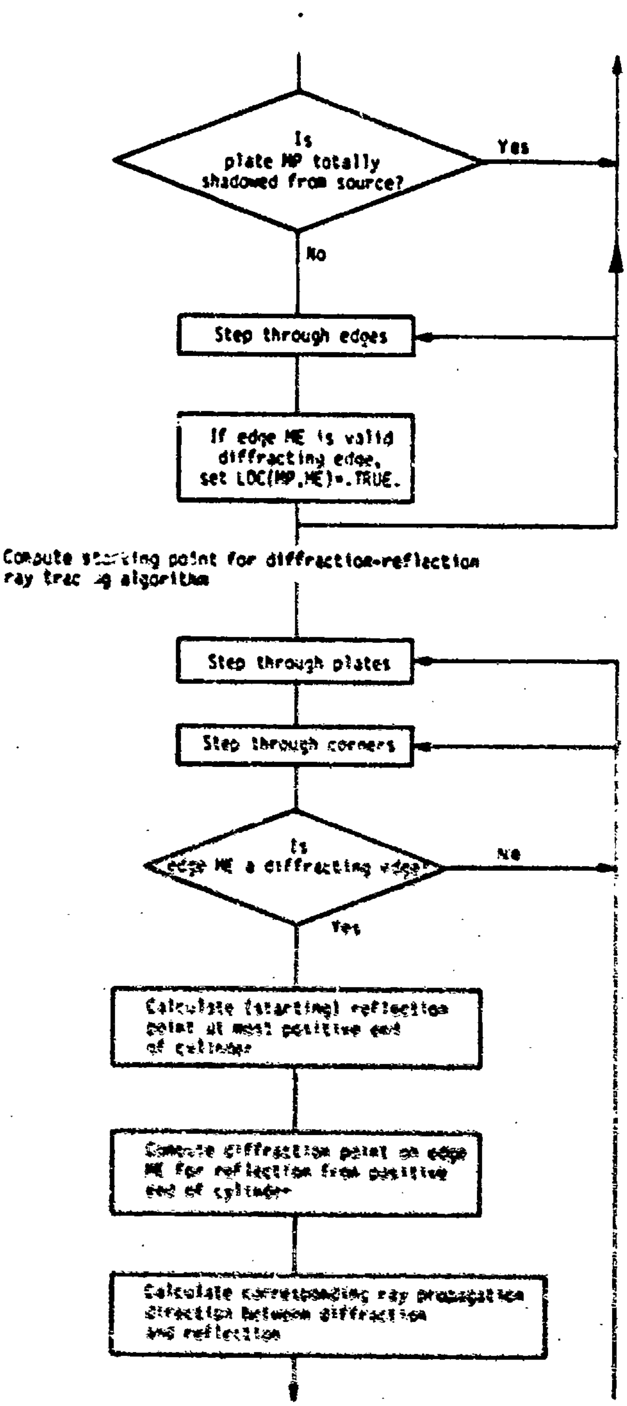
3. Determination of permissible range for cylinder reflected, plate diffracted term

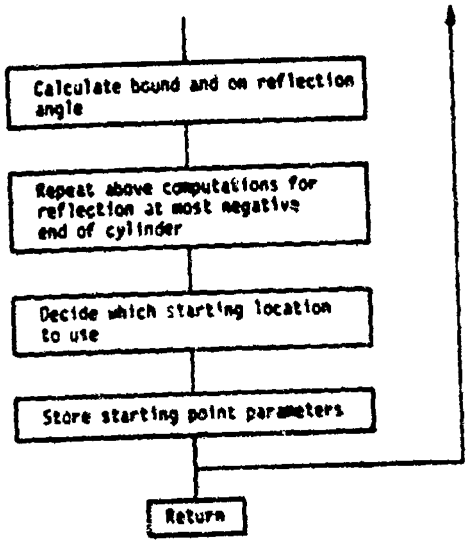



Step through edges
If corner ME is on cylinder, temporarily define new corner slightly off cylinder
Find reflection point on cylinder for source ray which reflects off cylinder and hits corner ME of plate MP
Take dot products of both edge vectors meeting at corner ME and ray incident on corner ME to obtain diffraction limit for cylinder-reflected, plate-diffracted ray
Step through plates
Step through edges
Compute branch cut displacement angle for diffracted-reflected and reflected-diffracted rays

4. Determination of parameters for ray diffracted by plate and then reflected off of cylinder




Is
plate MP totally
shadowed from source?
Yes
No
Step through edges
If edge ME is valid
diffracting edge,
set LOC(MP,ME)=.TRUE.
Compute starting point for diffraction-reflection
ray tracing algorithm
Step through plates
Step through corners
Is
edge ME a diffracting edge?
No
Yes
Calculate (starting) reflection
point at most positive end
of cylinder
Compute diffraction point on edge
ME for reflection from positive
end of cylinder
Calculate corresponding ray propagation
direction between diffraction
and reflection


Calculate bound and on reflection angle
Repeat above computations for reflection at most negative end of cylinder
Decide which starting location to use
Store starting point parameters
Return

## SYMBOL DICTIONARY

| | |
|---|---|
| BCD | DIFFRACTION LIMITS FOR RAY REFLECTED BY THE CYLINDER AND DIFFRACTED FROM PLATE |
| BTCN | X,Y COMPONENTS OF UNIT VECTORS FOR RAYS TANGENT TO THE CYLINDER FROM DIFFRACTION POINT ON PLATE EDGE (FOR MOST NEGATIVE STARTING POINT ON CYLINDER) |
| BTCP | X,Y COMPONENTS OF UNIT VECTORS FOR RAYS TANGENT TO THE CYLINDER FROM DIFFRACTION POINT ON PLATE EDGE (FOR MOST POSITIVE STARTING POINT ON CYLINDER) ALSO SEE BTI |
| BTDC | X,Y COMPONENTS OF UNIT VECTORS FOR RAYS TANGENT TO THE CYLINDER FROM DIFFRACTION POINT ON PLATE EDGE (FOR FAVORED STARTING POINT ON CYLINDER) |
| BTI | X AND Y COMPONENTS OF SOURCE IMAGE VECTORS TANGENT TO THE CYLINDER |
| DTCN | DOT PRODUCT OF UNIT VECTORS OF RAYS TANGENT TO CYLINDER FROM DIFFRACTION POINT (FOR MOST NEG. STARTING REFL POINT ON CYLINDER) |
| DTCP | DOT PRODUCT OF UNIT VECTORS OF RAYS TANGENT TO CYLINDER FROM DIFFRACTION POINT (FOR MOST POS STARTING REFL POINT ON CYLINDER) (ALSO SEE DTI) |
| DTDC | DOT PRODUCT OF UNIT VECTORS OF RAYS TANGENT TO CYLINDER FROM DIFFRACTION POINT (FOR FAVORED STARTING POINT ON CYLINDER) |
| DTI | DOT PRODUCT OF SOURCE IMAGE VECTORS TANGENT TO THE CYLINDER (SINGLE REFLECTION FROM PLATE MP) |
| LCD | SET TRUE IF CORNER ME OF PLATE MP IS ON CYLINDER |
| LDC | SET TRUE IF EDGE ME OF PLATE MP IS STRONG DIFFRACTING PART OF WEDGE (FNP<0) |
| MEC | INDEX VARIABLE USED TO DETERMINE COMMON EDGES |
| MEN | INDEX VARIABLE USED TO DETERMINE COMMON EDGES |
| MEX | MAXIMUM NUMBER OF EDGES ON PLATE MP |
| PDCR | PHI COMPONENT OF RAY PROPAGATION DIRECTION AFTER REFLECTION FROM CYLINDER (RAY DIFFRACTED BY PLATE EDGE AND THEN REFLECTED BY CYLINDER) |
| PHBR | BRANCH CUT DISPLACEMENT ANGLE FOR DIFFRACTION POINT ALONG EDGE ME OF PLATE MP |
| RC | DISTANCE FROM Z AXIS TO PLATE CORNER |
| RE | RADIUS OF CYLINDER AT POINT DEFINED BY ELL ANGLE VC |
| TDCR | THETA COMPONENT OF RAY PROPAGATION DIRECTION AFTER REFLECTION FROM CYLINDER (RAY DIFFRACTED BY PLATE EDGE AND THEN REFLECTED BY CYLINDER) |
| UCD | Z COMPONENT OF REFLECTION POINT LOCATION ON CYL. FOR RAY WHICH IS REFLECTED BY CYLINDER AND DIFFRACTED BY EDGE ME OF PLATE MP |
| UDC | Z COMPONENT OF STARTING POINT LOCATION ON CYLINDER (FOR RAY TRACING ALGORITHM) FOR RAY DIFFRACTED BY PLATE EDGE AND THEN REFLECTED BY CYLINDER |
| VC | ELLIPTIC ANGLE DEFINING LOCATION OF A CORNER (2-D) |
| VCD | ELL. ANGLE DEFINING REFLECTION POINT ON CYLINDER (2-D) FOR RAY WHICH IS REFLECTED BY CYLINDER AND DIFFRACTED BY EDGE ME OF PLATE MP |
| VDC | ELL ANGLE DEFINING STARTING POINT ON CYLINDER (FOR RAY-TRACING ALGORITHM) FOR RAY DIFFRACTED BY PLATE EDGE AND THEN REFLECTED BY CYLINDER |
| VI | X,Y,Z COMPONENTS OF PROPAGATION DIRECTION OF RAY INCIDENT ON CYLINDER REFLECTION POINT |
| VTI | ELL ANGLE DEFINING DIRECTION OF THE TWO RAYS FROM IMAGE SOURCE TANGENT TO THE CYLINDER (SINGLE REFL. OF SOURCE RAY FROM PLATE MP) |
| XC | MODIFIED PLATE CORNER LOCATION USED IN DETERMINING CYLINDER REFL, PLATE DIFFRACTION LIMITS |
| XDC | X,Y,Z COMPONENTS OF STARTING DIFF. POINT LOCATION ON EDGE ME (FOR RAY TRACING ALGORITHM) FOR RAY DIFF. BY PLATE EDGE AND REFLECTED BY CYLINDER |
| XCS | X,Y,Z COMPONENTS OF STARTING REFLECTION POINT ON CYL. |

# CODE LISTING

```
C---------------------------------------------------------------------
        SUBROUTINE GEOMPC
C!!!
C!!!    THIS SUBROUTINE COMPUTES ALL THE GEOMETRY ASSOCIATED
C!!!    WITH FIXED PARAMETERS FOR PLATE-CYLINDER INTERACTIONS
C!!!
        DIMENSION XII(3),XIN(3),VI(3),DS(3),XC(3),VNC(3)
        DIMENSION XOB(3),XDC(3),VTCP(2),BTCP(4),VTCN(2),BTCN(4)
        LOGICAL LPLA,LCYL,LDC,LCD(14,6),LSRFC,LSURF,LDEBUG,LTEST
        LOGICAL LIHD,LSHD,LSTD,LSTS,LCTD,LHCT,LHIT
        COMMON/PIS/PI,TPI,DPR,RPD
        COMMON/GEOMEL/A,B,ZC(2),SNC(2),CNC(2),CTC(2)
        COMMON/GEOPLA/X(14,6,3),V(14,6,3),VP(14,6,3),VN(14,3)
       2,MEP(14),MPX
        COMMON/EDMAG/VMAG(14,6)
        COMMON/SORINF/XS(3),VXS(3,3)
        COMMON/IMAINF/XI(14,14,3),VXI(3,3,14)
        COMMON/IMCINF/XIC(2,3),VXIC(3,3,2)
        COMMON/FARP/IM,H,HAW
        COMMON/SOURSF/FACTOR
        COMMON/ENDFCL/BD(14,6,2)
        COMMON/ENDSCL/DTS,VTS(2),BTS(4)
        COMMON/BNDICL/DTI(14),VTI(14,2),BTI(14,4)
        COMMON/BNDRCL/VCD(14,6),UCD(14,6),BCD(14,6,2)
        COMMON/ENDDCL/VDC(14,6),UDC(2),PDCR(14,6,2),TDCR(14,6,2)
       2,DTDC(14,6),BTDC(14,6,4),DDC(14,6,2)
        COMMON/SRFACC/LSRFC(2)
        COMMON/SURFAC/LSURF(14)
        COMMON/LPLCY/LPLA,LCYL
        COMMON/LSHDT/LSHD(14),LIHD(14,14)
        COMMON/LSHDP/LSTS,LSTD(14)
        COMMON/LDCBY/LDC(14,6)
        COMMON/TEST/LDEBUG,LTEST
        COMMON/FNANG/FNP(14,6)
        COMMON/ERNPHW/PHWR(14,6)
        IF(LDEBUG) WRITE(6,900)
  900   FORMAT(/,' DEBUGGING GEOMPC SUBROUTINE')
C!!!    1. DETERMINATION OF EDGES ATTACHED TO CYLINDER
        DO 3 MP=1,MPX
        MEX=MEP(MP)
        DO 3 ME=1,MEX
  3     LCD(MP,ME)=.FALSE.
C!!!    STEP THRU PLATES
        DO 17 MP=1,MPX
        MEX=MEP(MP)
        MEC=0
C!!!    STEP AROUND CORNERS ON PLATE MP
        DO 14 ME=1,MEX
        RC=X(MP,ME,1)*X(MP,ME,1)+X(MP,ME,2)*X(MP,ME,2)
        VC=BTAN2(A*X(MP,ME,2),B*X(MP,ME,1))
        XE=A*COS(VC)
        YE=B*SIN(VC)
        RE=XE*XE+YE*YE
        IF(ABS(RC-RE).GT.0.01) GO TO 14
        IF(X(MP,ME,3).GT.ZC(1)+XE*CTC(1).OR.
       2X(MP,ME,3).LT.ZC(2)+XE*CTC(2)) GO TO 14
        LCD(MP,ME)=.TRUE.
        X(MP,ME,1)=XE
        X(MP,ME,2)=YE
        IF(MEC.NE.0) GO TO 13
        MEC=ME
        GO TO 14
  13    MEN=MEC
        IF(ME-MEC.GT.1) MEN=MEX
C!!!    IF EDGE ME IS ATTACHED TO CYLINDER, SET WEDGE ANGLE INDICATOR
C!!!    TO -1 AS FLAG
```

```
      FNP(MP,MEN)=-1.
 14   CONTINUE
      IF(LDEBUG) WRITE(6,*) (LCD(MP,ME),ME=1,MEX)
 17   CONTINUE
C!!!  2. DETERMINATION OF IMAGE BOUNDS ON CYLINDER
C!!!  STEP THRU PLATES
      DO 62 MP=1,MPX
      DO 60 N=1,3
 60   XIN(N)=XI(MP,MP,N)
C!!!  CALCULATE TANGENT ANGLES AND UNIT VECTORS
      CALL TANG(DTCP,VTCP,BTCP,XIN)
      DTI(MP)=DTCP
      VTI(MP,1)=VTCP(1)
      VTI(MP,2)=VTCP(2)
      DO 61 J=1,4
 61   BTI(MP,J)=BTCP(J)
      IF(.NOT.LDEBUG) GO TO 62
      WRITE(6,*) DTI(MP)
      WRITE(6,*) VTI(MP,1),VTI(MP,2)
      WRITE(6,*) (BTI(MP,J),J=1,4)
 62   CONTINUE
C!!!  3. DETERMINATION OF PERMISSABLE RANGE FOR CYLINDER
C!!!  REFLECTED, PLATE DIFFRACTED FIELD
C!!!  INITIALIZE VALUES
      DO 90 MP=1,MPX
      MEX=MEP(MP)
      DO 90 ME=1,MEX
      VCD(MP,ME)=Ø.
      BCD(MP,ME,1)=Ø.
 90   BCD(MP,ME,2)=Ø.
C!!!  STEP THRU PLATES
      DO 92 MP=1,MPX
C!!!  IS PLATE MP TOTALLY SHADOWED FROM SOURCE?
      IF(LSHD(MP)) GO TO 92
      MEX=MEP(MP)
C!!!  STEP AROUND EDGES ON PLATE MP
      DO 91 ME=1,MEX
C!!!  IF EDGE ME IS ON CYLINDER, DEFINE NEW CORNER
C!!!  SLIGHTLY OFF CYLINDER
      IF(LCD(MP,ME)) GO TO 94
      DO 93 N=1,3
 93   XC(N)=X(MP,ME,N)
      GO TO 97
 94   VCR=BTAN2(X(MP,ME,2),X(MP,ME,1))
      SNX=B*COS(VCR)
      SNY=A*SIN(VCR)
      J=Ø
 95   J=J+1
      MJ=ME+1-J
      IF(MJ.EQ.Ø) MJ=MEX
      VCV=SNX*V(MP,MJ,1)+SNY*V(MP,MJ,2)
      IF(ABS(VCV).LT.1.E-5) GO TO 95
      SVCV=SIGN(.Ø1,VCV)
      DO 96 N=1,3
 96   XC(N)=X(MP,ME,N)+SVCV*V(MP,MJ,N)
 97   CONTINUE
C!!!  USE RAY TRACING TECHNIQUES TO DETERMINE REFL.
C!!!  POINT AND REFL. RAY DIRECTION OF SOURCE RAY REFL.
C!!!  FROM CYLINDER AND INCIDENT ON CORNER ME OF PLATE MP
C!!!  (SATISFY LAW OF REFLECTION)
      CALL RFDFIN(VR,UR,VI,XC)
      VCD(MP,ME)=VR
      UCD(MP,ME)=UR
      DO 91 J=1,2
      MJ=ME+1-J
      IF(MJ.EQ.Ø) MJ=MEX
      DO 91 N=1,3
C!!!  TAKE DOT PRODUCT OF RAY INCIDENT ON CORNER AND
C!!!  EDGE UNIT VECTOR TO OBTAIN DIFFRACTION LIMIT
91    BCD(MP,MJ,J)=BCD(MP,MJ,J)+V(MP,MJ,N)*VI(N)
      IF(.NOT.LDEBUG) GO TO 92
      WRITE(6,*) (VCD(MP,ME),ME=1,MEX)
      WRITE(6,*) (UCD(MP,ME),ME=1,MEX)
      WRITE(6,*) (BCD(MP,ME,1),BCD(MP,ME,2),ME=1,MEX)
92    CONTINUE
C!!!  DETERMINATION OF BRANCH CUT DISPLACEMENT ANGLE FOR
C!!!  REF-DIF AND DIF-REF TERMS
C!!!  STEP THRU PLATES
      DO 1Ø3 MP=1,MPX
      MEX=MEP(MP)
C!!!  STEP THRU EDGES
      DO 1Ø1 ME=1,MEX
      XPHW=X(MP,ME,1)+Ø.5*VMAG(MP,ME)*V(MP,ME,1)
      YPHW=X(MP,ME,2)+Ø.5*VMAG(MP,ME)*V(MP,ME,2)
      PHWR(MP,ME)=ETAN2(YPHW,XPHW)
1Ø1   CONTINUE
      IF(.NOT.LDEBUG) GO TO 1Ø3
      WRITE(6,*) (PHWR(MP,ME),ME=1,MEX)
1Ø3   CONTINUE
C!!!  4. DETERMINATION OF PARAMETERS FOR RAY DIFFRACTED
C!!!  BY PLATE EDGE AND THEN REFLECTED OFF OF CYLINDER
      DO 111 MP=1,MPX
      MEX=MEP(MP)
      DO 111 ME=1,MEX
111   LDC(MP,ME)=.FALSE.
C!!!  STEP THRU PLATES
      DO 114 MP=1,MPX
      IF(LSHD(MP)) GO TO 114
      MEX=MEP(MP)
C!!!  STEP THRU EDGES
      DO 113 ME=1,MEX
      IF(FNP(MP,ME).LT.Ø.) GO TO 112
      LDC(MP,ME)=.TRUE.
112   CONTINUE
113   CONTINUE
      IF(LDEBUG) WRITE(6,*) (LDC(MP,ME),ME=1,MEX)
114   CONTINUE
      UDC(1)=ZC(1)+A*CTC(1)
      UDC(2)=ZC(2)+A*CTC(2)
      IF(LDEBUG) WRITE(6,*) UDC(1),UDC(2)
C!!!  STEP THRU PLATES
      DO 118 MP=1,MPX
      MEX=MEP(MP)
C!!!  STEP THRU CORNERS
      DO 118 ME=1,MEX
      IF(.NOT.LDC(MP,ME)) GO TO 118
      MJ=ME+1
      IF(MJ.GT.MEX) MJ=1
      VDCA=BTAN2(A*X(MP,ME,2),B*X(MP,ME,1))
      VDCB=BTAN2(A*X(MP,MJ,2),B*X(MP,MJ,1))
      VDC(MP,ME)=.5*(VDCA+VDCB)
C!!!  CALCULATE (STARTING) REFLECTION POINT AT MOST
C!!!  POSITIVE END OF CYLINDER
      XOB(1)=A*COS(VDC(MP,ME))
      XOB(2)=B*SIN(VDC(MP,ME))
      XOB(3)=UDC(1)
      VNX=B*COS(VDC(MP,ME))
      VNY=A*SIN(VDC(MP,ME))
C!!!  COMPUTE STARTING DIFFRACTION POINT CORRESPONDING
C!!!  TO REFLECTION POINT AT MOST POS. END OF CYL.
      CALL DPTNEW(XS,XOB,XDC,ME,MP)
C!!!  CALCULATE CORRESPONDING REFLECTED RAY PROPAGATION
```

```
C!!!  DIRECTION FOR ABOVE DIFFRACTION AND REFL. POINTS
      VI(1)=XOB(1)-XDC(1)
      VI(2)=XOB(2)-XDC(2)
      VI(3)=XOB(3)-XDC(3)
      CNIP=VNX*VI(1)+VNY*VI(2)
      CNIN=VNX*VI(2)-VNY*VI(1)
      PDCR(MP,ME,1)=BTAN2((CNIN*VNX-CNIP*VNY),-(CNIP*VNX+CNIN*VNY))
      CPDC=COS(PDCR(MP,ME,1))
      SPDC=SIN(PDCR(MP,ME,1))
      TDCR(MP,ME,1)=BTAN2(-CNIP,(VNX*CPDC+VNY*SPDC)*VI(3))
C!!!  CALCULATE BOUND ON REFLECTION ANGLE
      DDC(MP,ME,1)=COS(TDCR(MP,ME,1))
C!!!  REPEAT CALCULATIONS FOR MOST NEGATIVE ENDCAP
      CALL TANG(DTCP,VTCP,BTCP,XDC)
      XOB(3)=UDC(2)
      CALL DP1NFW(XS,XOB,XDC,ME,MP)
      VI(1)=XOB(1)-XDC(1)
      VI(2)=XOB(2)-XDC(2)
      VI(3)=XOB(3)-XDC(3)
      CNIP=VNX*VI(1)+VNY*VI(2)
      CNIN=VNX*VI(2)-VNY*VI(1)
      PDCR(MP,ME,2)=BTAN2((CNIN*VNX-CNIP*VNY),-(CNIP*VNX+CNIN*VNY))
      CPDC=COS(PDCR(MP,ME,2))
      SPDC=SIN(PDCR(MP,ME,2))
      TDCR(MP,ME,2)=BTAN2(-CNIP,(VNX*CPDC+VNY*SPDC)*VI(3))
      DDC(MP,ME,2)=COS(TDCR(MP,ME,2))
      CALL TANG(DTCN,VTCN,BTCN,XDC)
C!!!  DECIDE WHICH STARTING LOCATION TO USE BETWEEN MOST
C!!!  NEGATIVE AND MOST POS. END CAP VALUES
      IF(DTCN.GT.DTCP) GO TO 116
      DTDC(MP,ME)=DTCP
      DO 115 J=1,4
  115 BTDC(MP,ME,J)=BTCP(J)
      GO TO 119
  116 DTDC(MP,ME)=DTCN
      DO 117 J=1,4
  117 BTDC(MP,ME,J)=BTCN(J)
  119 CONTINUE
      IF(.NOT.LDEBUG) GO TO 118
      WRITE(6,*) VDC(MP,ME),(PDCR(MP,ME,J),TDCR(MP,ME,J),J=1,2)
     2,DTDC(MP,ME)
      WRITE(6,*) (BTDC(MP,ME,J),J=1,4),(DDC(MP,ME,J),J=1,2)
  118 CONTINUE
      RETURN
      END
```

## IMAGE

PURPOSE

To determine location of source image for reflection of source ray off of plate MP. (double reflection image locations may be obtained by calling IMAGE twice; once for the source ray reflection from plate MP and once for the reflection of the ray from the image location off of the second plate.)

PERTINENT GEOMETRY

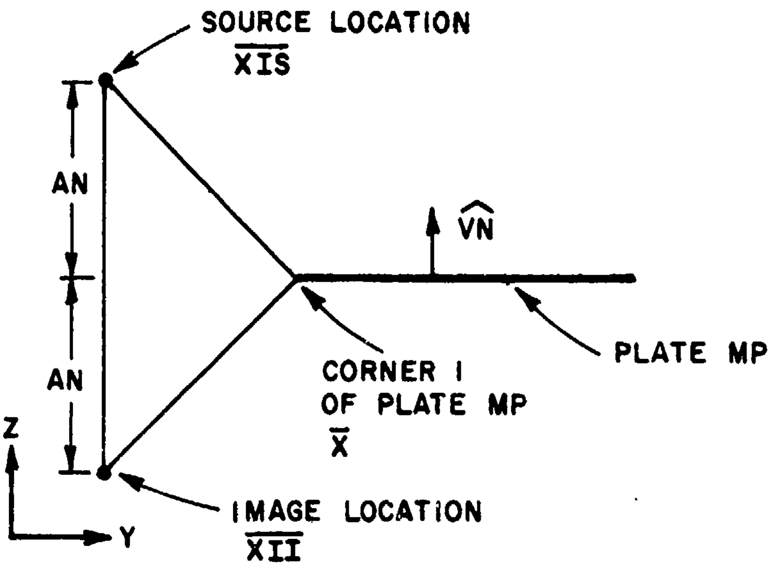


Figure 75.-- Geometry for determining source image location.

$\widehat{VN}$ = plate unit normal = $\hat{x}$ VN(MP,1) + $\hat{y}$ VN(MP,2) + $\hat{z}$ VN(MP,3)

$\overline{XIS}$ = $\hat{x}$ XIS(1) + $\hat{y}$ XIS(2) + $\hat{z}$ XIS(3)

$\overline{X}$ = $\hat{x}$ X(MP,1,1) + $\hat{y}$ X(MP,1,2) + $\hat{z}$ X(MP,1,3)

METHOD

The source image location is given by

$\overline{XII}$ = $\overline{XIS}$ + 2 AN $\widehat{VN}$ = $\hat{x}$ XII(1) + $\hat{y}$ XII(2) + $\hat{z}$ XII(3)

where

AN = $(\overline{X} - \overline{XIS})\cdot\widehat{VN}$

and $\overline{X}$, $\overline{XIS}$, and $\widehat{VN}$ are as shown in Figure 75.

## FLOW DIAGRAM

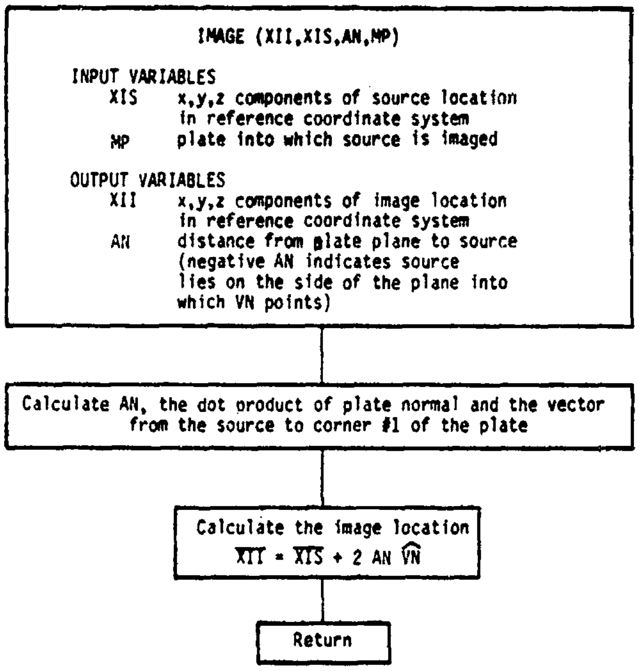


IMAGE (XII,XIS,AN,MP)

INPUT VARIABLES
XIS x,y,z components of source location in reference coordinate system
MP plate into which source is imaged

OUTPUT VARIABLES
XII x,y,z components of image location in reference coordinate system
AN distance from plate plane to source (negative AN indicates source lies on the side of the plane into which VN points)

Calculate AN, the dot oroduct of plate normal and the vector from the source to corner #1 of the plate

Calculate the image location
$\overline{XII} = \overline{XIS} + 2\ AN\ \widehat{VN}$

Return

## SYMBOL DICTIONARY

| | |
|---|---|
| AN | DOT PRODUCT OF VECTOR FROM SOURCE TO EDGE ONE OF PLATE MP AND THE PLATE UNIT NORMAL |
| MP | PLATE INTO WHICH SOURCE IS IMAGED |
| XII | X,Y,Z COMPONENTS OF IMAGE LOCATION IN RCS |
| XIS | X,Y,Z COMPONENTS OF SOURCE LOCATION IN RCS |

## CODE LISTING

```
C-----------------------------------------------------------------------
      SUBROUTINE IMAGE(XII,XIS,AN,MP)
C!!!
C!!!  DETERMINE IMAGE POSITION FOR SOURCE XIS IN PLATE #MP.
C!!!  AN INDICATES WHICH SIDE OF PLATE SOURCE IS LOCATED
C!!!  RELATIVE TO PLATE NORMAL.
C!!!
      DIMENSION XII(3),XIS(3)
      COMMON/GEOPLA/X(14,6,3),V(14,6,3),VP(14,6,3),VN(14,3)
     2,MEP(14),MPX
      AN=0.
      DO 10 N=1,3
   10 AN=AN+(X(MP,1,N)-XIS(N))*VN(MP,N)
      DO 20 N=1,3
   20 XII(N)=XIS(N)+2.*AN*VN(MP,N)
      RETURN
      END
```

## IMCDIR

### PURPOSE

To determine the source image axes directions for a source after reflection off a given end cap.

### PERTINENT GEOMETRY

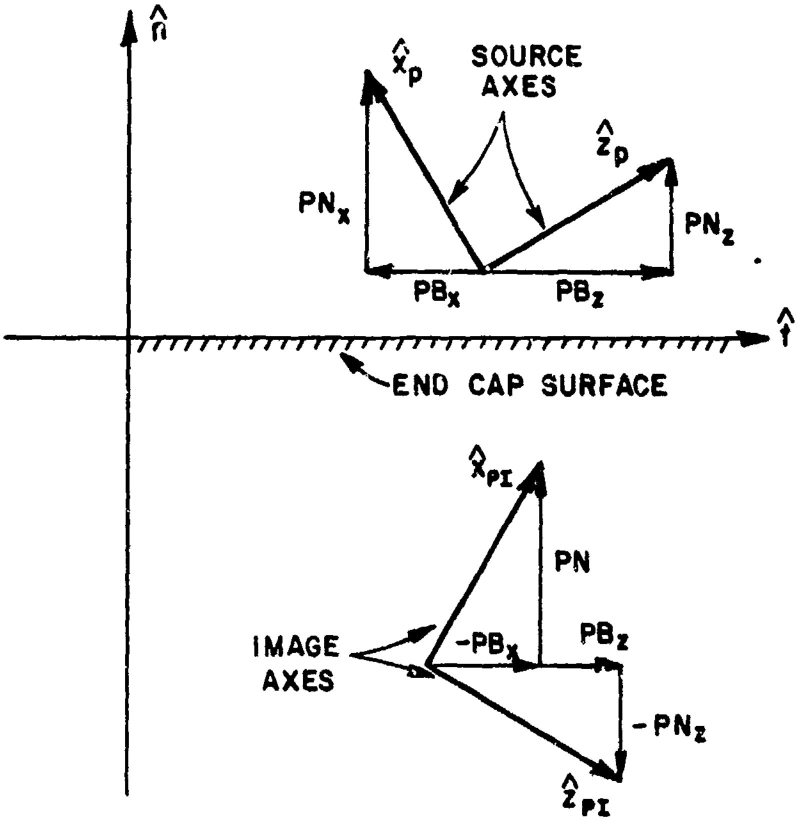


Figure 76a-- Illustration of imaging of source axes for magnetic source.

$\hat{x}_p$ = $\hat{x}$ VSOURC(1,1) + $\hat{y}$ VSOURC(1,2) + $\hat{z}$ VSOURC(1,3)

$\hat{x}_{pi}$ = $\hat{x}$ VIMAG(1,1) + $\hat{y}$ VIMAG(1,2) + $\hat{z}$ VIMAG(1,3)

$\hat{n}$ = unit normal of endcap

$\hat{t}$ = unit vector tangent to endcap

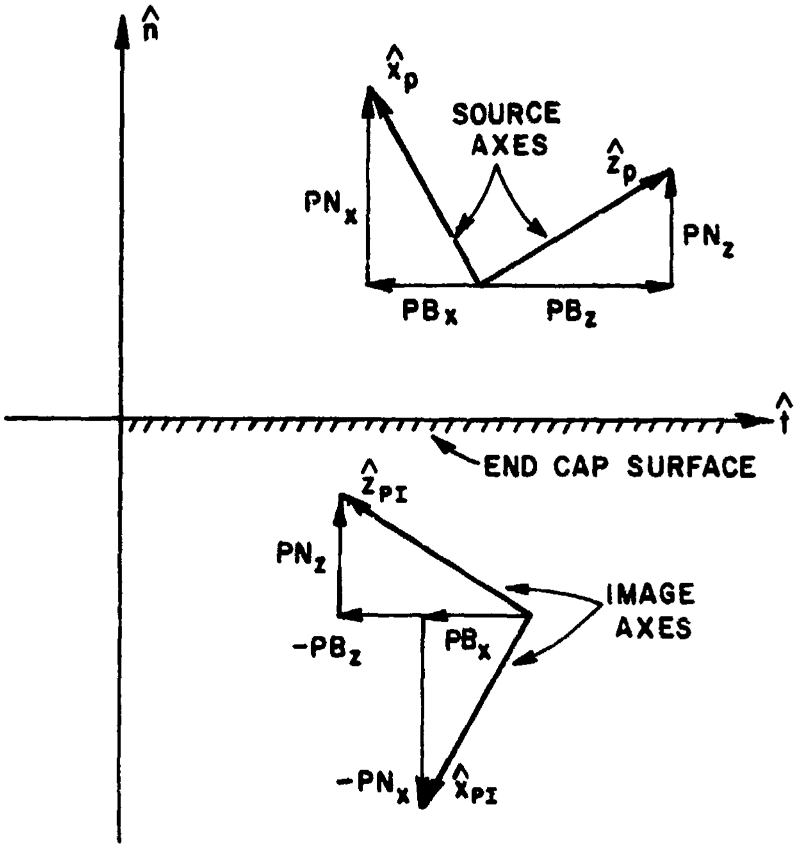


Figure 76b--Imaging of source axes for electric source.

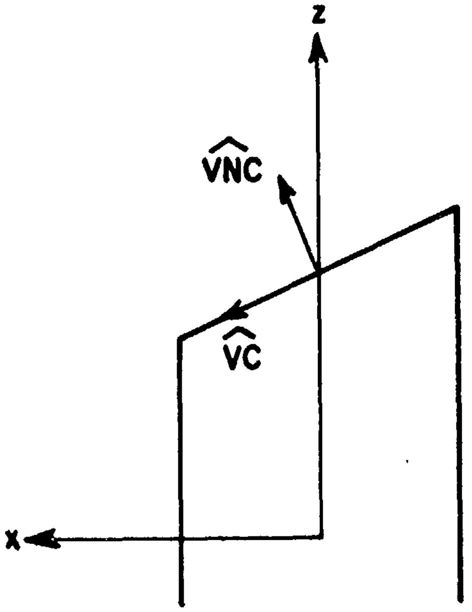


Figure 77--Endcap coordinate system.

$$\hat{n} = \widehat{VNC} = \hat{x}\ VNC(1) + \hat{y}\ VNC(2) + \hat{z}\ VNC(3), \quad VNC(2)=0$$

$$\hat{t} = \widehat{VC} = \hat{x}\ VC(1) + \hat{y}\ VC(2) + \hat{z}\ VC(3) \quad , \quad VC(2)=0$$

$$\hat{b} = \hat{y}$$

METHOD

The source image axes unit vectors for an electric source imaged in an end cap are given by

$$\hat{x}_{pi} = (-\hat{x}_p \cdot \hat{n})\hat{n} + (\hat{x}_p \cdot \hat{t})\hat{t} + (\hat{x}_p \cdot \hat{b})\hat{b}$$

$$\hat{z}_{pi} = (\hat{z}_p \cdot \hat{n})\hat{n} + (-\hat{z}_p \cdot \hat{t})\hat{t} + (-\hat{z}_p \cdot \hat{b})\hat{b}$$

$$\hat{y}_{pi} = \hat{z}_{pi} \times \hat{x}_{pi}$$

For a magnetic source, the axes are given by

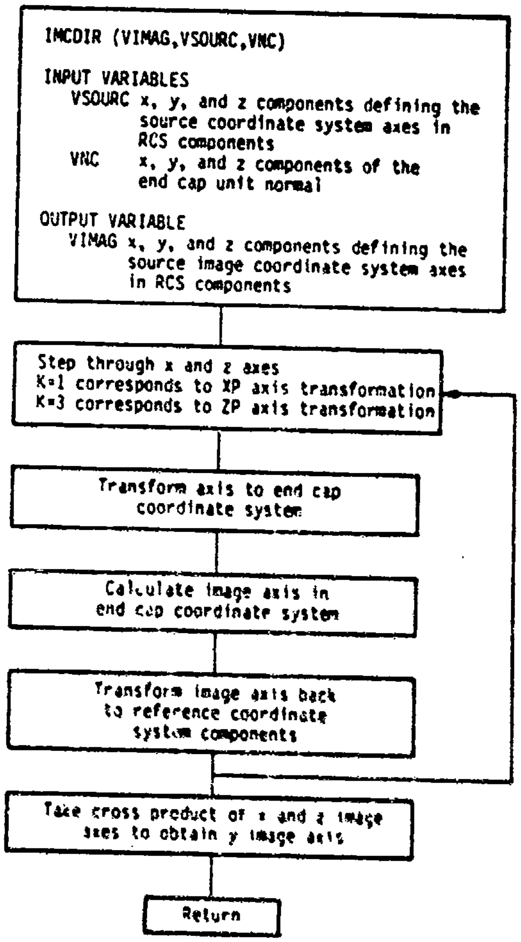
$$\hat{x}_{pi} = (\hat{x}_p \cdot \hat{n})\hat{n} + (-\hat{x}_p \cdot \hat{t})\hat{t} + (-\hat{x}_p \cdot \hat{b})\hat{b}$$

$$\hat{z}_{pi} = (-\hat{z}_p \cdot \hat{n})\hat{n} + (\hat{z}_p \cdot \hat{t})\hat{t} + (\hat{z}_p \cdot \hat{b})\hat{b}$$

$$\hat{y}_{pi} = \hat{z}_{pi} \times \hat{x}_{pi}$$

where $\hat{x}_p$, $\hat{y}_p$, $\hat{z}_p$ are unit vectors of the source coordinate system axes and $\hat{x}_{pi}$, $\hat{y}_{pi}$, $\hat{z}_{pi}$ are the unit vectors of the source image coordinate system for the end cap.

## FLOW DIAGRAM


IMCDIR (VIMAG,VSOURC,VNC)
INPUT VARIABLES
VSOURC x, y, and z components defining the source coordinate system axes in RCS components
VNC x, y, and z components of the end cap unit normal
OUTPUT VARIABLE
VIMAG x, y, and z components defining the source image coordinate system axes in RCS components
Step through x and z axes
K=1 corresponds to XP axis transformation
K=3 corresponds to ZP axis transformation
Transform axis to end cap coordinate system
Calculate image axis in end cap coordinate system
Transform image axis back to reference coordinate system components
Take cross product of x and z image axes to obtain y image axis
Return

## SYMBOL DICTIONARY

| | |
|---|---|
| K | INDEX VARIABLE |
| L | INDEX VARIABLE |
| LL | INDEX VARIABLE |
| PB | DOT PRODUCT OF END CAP UNIT BINORMAL AND UNIT VECTOR OF SOURCE AXIS BEING IMAGED |
| PN | DOT PRODUCT OF END CAP UNIT NORMAL AND UNIT VECTOR OF SOURCE AXIS BEING IMAGED |
| PT | DOT PRODUCT OF END CAP UNIT TANGENT AND UNIT VECTOR OF SOURCE AXIS BEING IMAGED |
| VC | X,Y,Z COMPONENTS OF UNIT VECTOR TANGENT TO END CAP (IN X-Z PLANE) |
| VIMAG | ARRAY OF COMPONENTS DEFINING THE SOURCE IMAGE COORDINATE SYSTEM AXES IN (X,Y,Z) REF COORD SYSTEM COMPONENTS |
| VNC | X,Y, AND Z COMPONENTS OF END CAP UNIT NORMAL |
| VSOURC | ARRAY OF COMPONENTS DEFINING THE SOURCE COORDINATE SYSTEM AXES IN (X,Y,Z) REFERENCE COORD SYS. COMPONENTS |
| VX<br>VY<br>VZ | X,Y, AND Z COMPONENTS OF SOURCE AXIS BEING IMAGED |

## CODE LISTING

```
C-----------------------------------------------------------------
      SUBROUTINE IMCDIR(VIMAG,VSOURC,VNC)
C!!!
C!!!  DETERMINES DIRECTION OF IMAGE SOURCE COORDINATE
C!!!  SYSTEM FOR THE END CAPS.
C!!!
      DIMENSION VIMAG(3,3),VSOURC(3,3),VNC(3),VC(3)
      COMMON/PARP/IM,H,HAM
      VC(1)=VNC(3)
      VC(2)=0.
      VC(3)=-VNC(1)
C!!!  IMAGE X AND Z DIPOLE AXES
      DO 15 LL=1,2
      L=LL-1
      K=1+2*L
C!!!  TRANSFORM AXIS TO END CAP COORDINATE SYSTEM
      VX=VSOURC(K,1)
      VY=VSOURC(K,2)
      VZ=VSOURC(K,3)
      PN=VX*VNC(1)+VY*VNC(2)+VZ*VNC(3)
      PT=VX*VC(1)+VY*VC(2)+VZ*VC(3)
      PB=VY
C!!!  FIND IMAGE AXIS
      IF((IM+L).EQ.1) GO TO 10
      PN=-PN
      GO TO 20
   10 PB=-PB
      PT=-PT
   20 CONTINUE
C!!!  TRANSFORM IMAGE AXIS BACK TO REFERENCE COORDINATE SYSTEM
      VIMAG(K,1)=PN*VNC(1)+PT*VC(1)
      VIMAG(K,2)=PN*VNC(2)+PT*VC(2)+PB
      VIMAG(K,3)=PN*VNC(3)+PT*VC(3)
   15 CONTINUE
C!!!  TAKE CROSS PRODUCT OF Z AND X IMAGE AXES TO
C!!!  OBTAIN Y IMAGE AXIS
      VIMAG(2,1)=VIMAG(3,2)*VIMAG(1,3)-VIMAG(3,3)*VIMAG(1,2)
      VIMAG(2,2)=VIMAG(3,3)*VIMAG(1,1)-VIMAG(3,1)*VIMAG(1,3)
      VIMAG(2,3)=VIMAG(3,1)*VIMAG(1,2)-VIMAG(3,2)*VIMAG(1,1)
      RETURN
      END
```

## IMDIR

PURPOSE

To determine the image source axes directions for a source (or source image) after reflection off of a given plate.

PERTINENT GEOMETRY

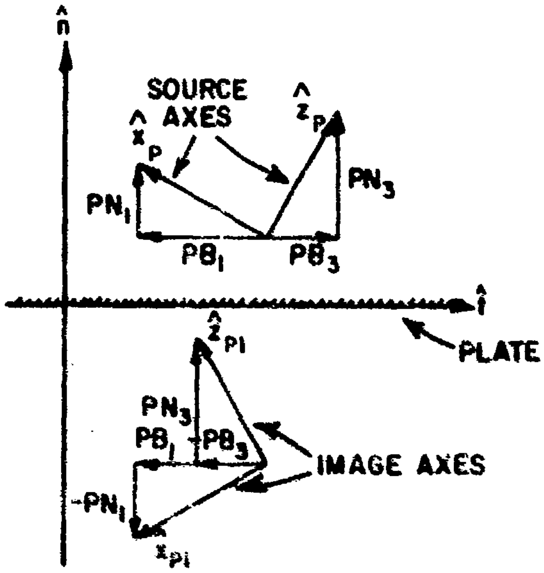


Figure 78a--Imaging of source coordinate system for electric source (shown in two dimensions for simplicity).

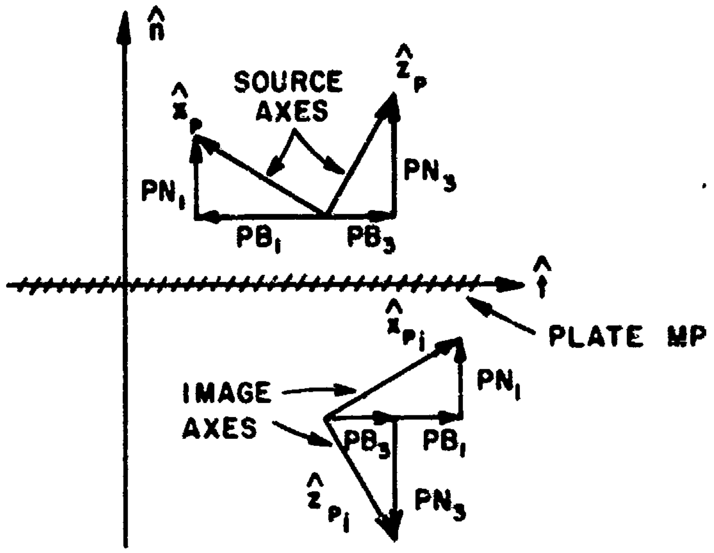


Figure 78b--Imaging of source coordinate system for magnetic source (shown in two dimensions for simplicity)

The current flows in the $\hat{z}_p$ direction.

$\hat{n}$ = plate unit normal = $\hat{x}$ VN(MP,1) + $\hat{y}$ VN(MP,2) + $\hat{z}$ VN(MP,3)

$\hat{t}$ = unit vector tangent to plate = $\hat{x}$ V(MP,1,1) + $\hat{y}$ V(MP,1,2) + $\hat{z}$ V(MP,1,3)

$\hat{b} = \hat{n} \times \hat{t}$ = $\hat{x}$ VP(MP,1,1) + $\hat{y}$ VP(MP,1,2) + $\hat{z}$ VP(MP,1,3)

(unit vectors $\hat{t}$ and $\hat{b}$ arbitrarily chosen to be edge vector $\hat{V}$ and binormal $\hat{VP}$ of edge #1 on the plate).

METHOD

The source image axes unit vectors for an electric source are given by

$$\hat{x}_{pi} = (-\hat{x}_p \cdot \hat{n})\hat{n} + (\hat{x}_p \cdot \hat{t})\hat{t} + (\hat{x}_p \cdot \hat{b})\hat{b}$$
$$\hat{z}_{pi} = (\hat{z}_p \cdot \hat{n})\hat{n} + (-\hat{z}_p \cdot \hat{t})\hat{t} + (-\hat{z}_p \cdot \hat{b})\hat{b}$$
$$\hat{y}_{pi} = \hat{z}_{pi} \times \hat{x}_{pi}.$$

For a magnetic source the axes are given by

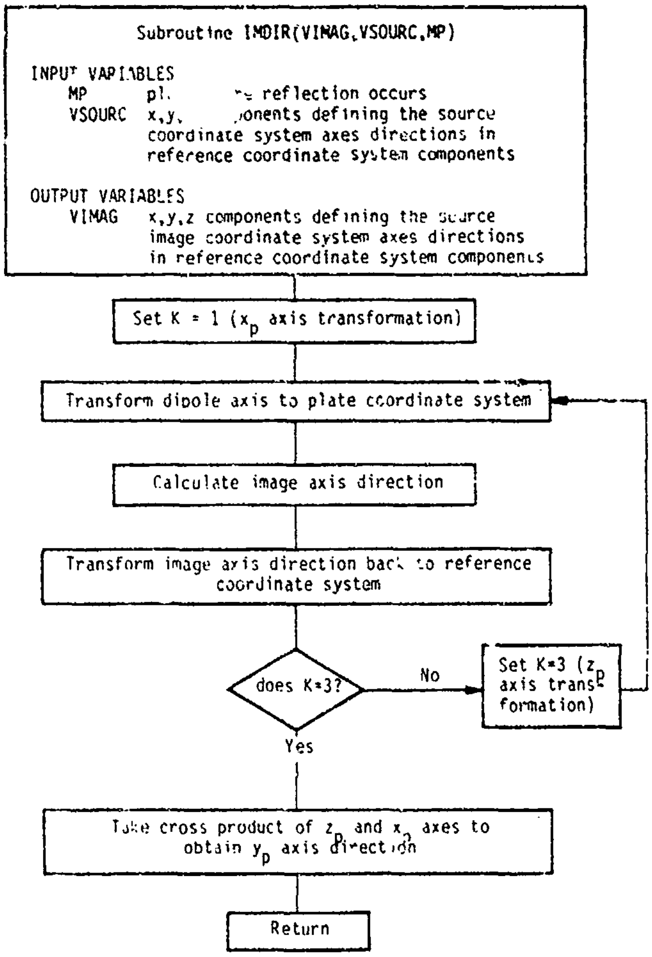
$$\hat{x}_{pi} = (\hat{x}_p \cdot \hat{n})\hat{n} + (-\hat{x}_p \cdot \hat{t})\hat{t} + (-\hat{x}_p \cdot \hat{b})\hat{b}$$

$$\hat{z}_{pi} = (-\hat{z}_p \cdot \hat{n})\hat{n} + (\hat{z}_p \cdot \hat{t})\hat{t} + (\hat{z}_p \cdot \hat{b})\hat{b}$$

$$\hat{y}_{pi} = \hat{z}_{pi} \times \hat{x}_{pi},$$

where $\hat{x}_p$, $\hat{y}_p$, $\hat{z}_p$ are the unit vectors of the source coordinate system axes and $\hat{x}_{pi}$, $\hat{y}_{pi}$, $\hat{z}_{pi}$ are the unit vectors of the source image coordinate system.

FLOW DIAGRAM


Subroutine IMDIR(VIMAG,VSOURC,MP)
INPUT VARIABLES
MP pl[illegible] re reflection occurs
VSOURC x,y,[illegible] ponents defining the source coordinate system axes directions in reference coordinate system components
OUTPUT VARIABLES
VIMAG x,y,z components defining the source image coordinate system axes directions in reference coordinate system components
Set K = 1 (x_p axis transformation)
Transform dipole axis to plate coordinate system
Calculate image axis direction
Transform image axis direction back to reference coordinate system
does K=3?
No
Set K=3 (z_p axis transformation)
Yes
Take cross product of z_p and x_p axes to obtain y_p axis direction
Return

## SYMBOL DICTIONARY

| | |
|---|---|
| K | K=1 CORRESPONDS TO XP AXIS TRANSFORMATION,<br>K=3 CORRESPONDS TO ZP AXIS TRANSFORMATION |
| L | INCREMENTAL VARIABLE |
| MP | PLATE OF REFLECTION |
| PB | COMPONENT OF AXIS IN PLATE PLANE NORMAL TO EDGE |
| PN | COMPONENT OF AXIS NORMAL TO PLATE |
| PT | COMPONENT OF AXIS PARALLEL TO PLATE EDGE |
| VIMAG | X,Y,Z COMPONENTS DEFINING UNIT VECTORS OF THE IMAGE SOURCE COORDINATE SYSTEM AXES IN RCS |
| VSOURC | X,Y,Z COMPONENTS DEFINING UNIT VECTORS OF THE SOURCE COORDINATE SYSTEM AXES IN RCS |
| VX<br>VY<br>VZ | X,Y, AND Z COMPONENTS OF AXIS UNDER TRANSFORMATION IN RCS |

## CODE LISTING

```
C-------------------------------------------------------------------------
      SUBROUTINE IMDIR(VIMAG,VSOURC,MP)
C!!!
C!!!  DETERMINES DIRECTION OF IMAGE SOURCE COORDINATE
C!!!  SYSTEM FOR PLATE #MP.
      DIMENSION VIMAG(3,3),VSOURC(3,3)
      COMMON/FARP/IM,H,HAW
      COMMON/GEOPLA/X(14,6,3),V(14,6,3),VP(14,6,3),VN(14,3)
     2,MEP(14),MPX
C!!!  IMAGE X AND Z DIPOLE AXES
      DO 15 LL=1,2
      L=LL-1
      K=1+2*L
C!!!  TRANSFORM AXIS TO PLATE COORDINATE SYSTEM
      VX=VSOURC(K,1)
      VY=VSOURC(K,2)
      VZ=VSOURC(K,3)
      PN=VX*VN(MP,1)+VY*VN(MP,2)+VZ*VN(MP,3)
      PT=VX*V(MP,1,1)+VY*V(MP,1,2)+VZ*V(MP,1,3)
      PB=VX*VP(MP,1,1)+VY*VP(MP,1,2)+VZ*VP(MP,1,3)
C!!!  FIND IMAGE AXIS
      IF((IM+L).EQ.1) GO TO 10
      PN=-PN
      GO TO 20
   10 PB=-PB
      PT=-PT
   20 CONTINUE
C!!!  TRANSFORM IMAGE AXIS BACK TO REFERENCE COORDINATE SYSTEM
      DO 15 N=1,3
   15 VIMAG(K,N)=PN*VN(MP,N)+PT*V(MP,1,N)+PB*VP(MP,1,N)
C!!!  TAKE CROSS PRODUCT OF Z AND X AXES TO OBTAIN Y AXIS
      VIMAG(2,1)=VIMAG(3,2)*VIMAG(1,3)-VIMAG(3,3)*VIMAG(1,2)
      VIMAG(2,2)=VIMAG(3,3)*VIMAG(1,1)-VIMAG(3,1)*VIMAG(1,3)
      VIMAG(2,3)=VIMAG(3,1)*VIMAG(1,2)-VIMAG(3,2)*VIMAG(1,1)
      RETURN
      END
```

## INCFLD

### PURPOSE

To calculate the far-zone electric field transmitted by the source in a given direction with phase referred to the reference coordinate system origin.

### PERTINENT GEOMETRY

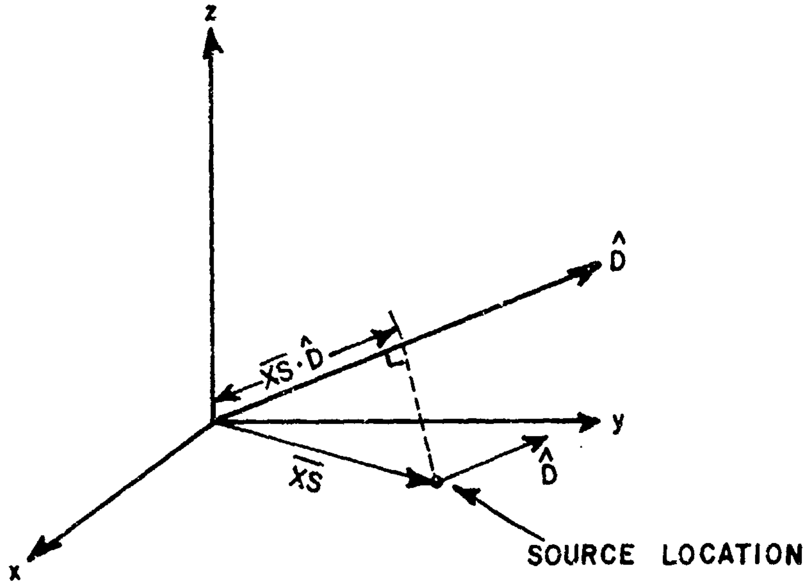


Figure 79--Geometry for source radiated field.

$\overline{XS}$ = source location = $\hat{x}$ XS(1) + $\hat{y}$ XS(2) + $\hat{z}$ XS(3)

$\hat{D}$ = propagation direction unit vector = $\hat{x}$ D(1) + $\hat{y}$ D(2) + $\hat{z}$ D(3)

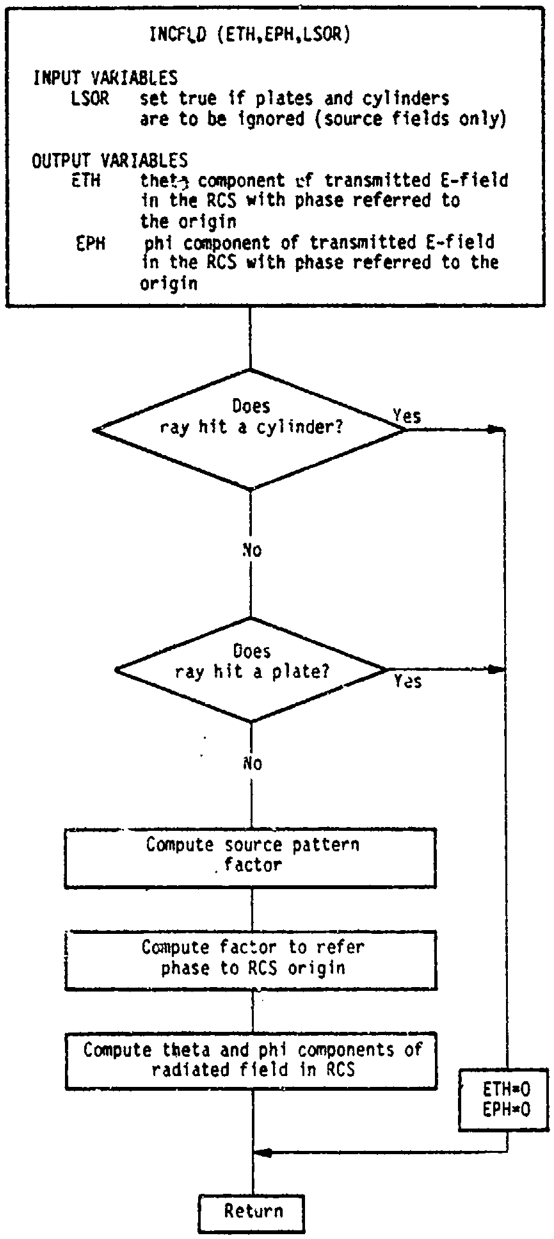
### METHOD

The direct field from the source incident upon the far zone observation point is found by adding the far field phase factor $e^{jk\hat{D}\cdot\overline{XS}}$ to the source pattern factor. The existance of the field is first tested by checking if the ray from the source to the observer is shadowed by a plate or cylind'". If it is shadowed the field is set to zero. If it is not shadowed the field is given by

$$\overline{E}^i(r,\theta,\phi) = W_m(ETH\hat{\theta}+EPH\hat{\phi})\,\frac{e^{-jkR}}{R}\ .$$

The factor $\frac{e^{-jkR}}{R}$ and source weight ($W_m$) are added elsewhere in the code.

## FLOW DIAGRAM


INCFLD (ETH,EPH,LSOR)
INPUT VARIABLES
LSOR set true if plates and cylinders are to be ignored (source fields only)
OUTPUT VARIABLES
ETH theta component of transmitted E-field in the RCS with phase referred to the origin
EPH phi component of transmitted E-field in the RCS with phase referred to the origin
Does ray hit a cylinder?
Yes
No
Does ray hit a plate?
Yes
No
Compute source pattern factor
Compute factor to refer phase to RCS origin
Compute theta and phi components of radiated field in RCS
ETH=0
EPH=0
Return

## SYMBOL DICTIONARY

| | |
|---|---|
| D | X,Y,Z COMPONENTS OF RAY PROPAGATION DIRECTION IN RCS |
| EPH | E-PHI COMPONENT OF SOURCE FIELD |
| ETH | E-THETA COMPONENT OF SOURCE FIELD |
| LHIT | SET TRUE IF RAY HITS PLATE OR CYLINDER |
| PH | COMPLEX PHASE CONSTANT (USED TO REFER PHASE TO RCS ORIGIN) |
| PHSR | PHI COMPONENT OF PROPAGATION DIRECTION IN RCS |
| THSR | THETA COMPONENT OF PROPAGATION DIRECTION IN RCS |

## CODE LISTING

```
C-----------------------------------------------------------------------
      SUBROUTINE INCFLD(ETH,EPH,LSOR)
C!!!
C!!!  COMPUTES THE DIRECT FIELD FROM THE SOURCE WITH PHASE
C!!!  REFERRED TO THE ORIGIN.
C!!!
      COMPLEX ETH,EPH,PH,CJ,CPI4,EX,EY,EZ
      LOGICAL LSOR,LHIT
      COMMON/SORINF/XS(3),VXS(3,3)
      COMMON/PIS/PI,TPI,DPR,RPD
      COMMON/DIR/D(3),THSR,PHSR,SPS,CPS,STHS,CTHS
      COMMON/COMP/CJ,CPI4
      COMMON/THPHUV/DT(3),DP(2)
      IF(LSOR)GO TO 1
C!!!  DOES RAY HIT A CYLINDER?
      CALL CYLINT(XS,D,PHSR,DHIT,LHIT,.FALSE.)
      IF(LHIT) GO TO 12
C!!!  DOES RAY HIT A PLATE?
      CALL PLAINT(XS,D,DHIT,0,LHIT)
      IF(LHIT) GO TO 12
C!!!  IF RAY DOES NOT HIT ANYTHING, COMPUTE SOURCE FIELD
C!!!  PATTERN FACTOR
 1    CALL SOURCE(ETH,EPH,EX,EY,EZ,THSR,PHSR,VXS)
C!!!  COMPUTE PHASE FACTOR
      PH=CEXP(CJ*TPI*(XS(1)*D(1)+XS(2)*D(2)+XS(3)*D(3)))
C!!!  COMPUTE THETA AND PHI COMPONENTS OF RADIATED
C!!!  FIELD IN RCS
      ETH=PH*ETH
      EPH=PH*EPH
      RETURN
 12   ETH=(0.,0.)
      EPH=(0.,0.)
      RETURN
      END
```

## NANDB

PURPOSE

To calculate the unit vectors for rays normal and tangent to the elliptical cylinder at a given point $\overline{XC}$ (in x-y plane) defined by elliptic angle VR.

PERTINENT GEOMETRY

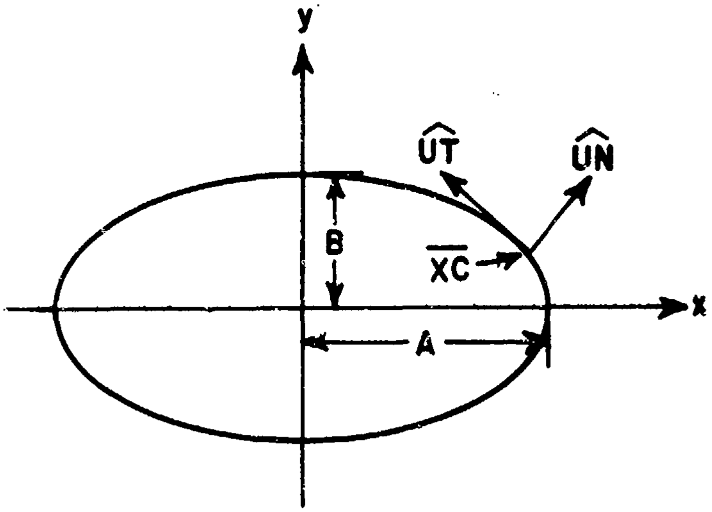


Figure 80--Illustration of unit vectors tangent and normal to the cylinder.

$\widehat{UT} = \hat{x}\ UT(1) + \hat{y}\ UT(2)$

$\widehat{UN} = \hat{x}\ UN(1) + \hat{y}\ UN(2)$

$\overline{XC} = \hat{x}\ A \cos(VR) + \hat{y}\ B \sin(VR)$

## METHOD

For the point on the cylinder defined by the elliptic angle VR, the unit normal vector is given as

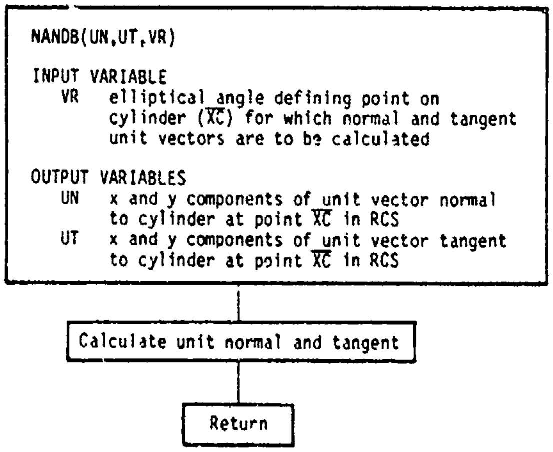
$$\hat{UN} = \frac{\hat{x}\ B\ \cos(VR)\ +\ \hat{y}\ A\ SIN(VR)}{\sqrt{B^2\ \cos^2(VR)\ +\ A^2\ \sin^2(VR)}}$$

and the unit tangent vector is given by

$$\hat{UT} = \frac{-\hat{x}\ A\ \sin(VR)\ +\ \hat{y}\ B\ \cos(VR)}{\sqrt{B^2\ \cos^2(VR)\ +\ A^2\ \sin^2(VR)}}$$

as shown in Figure 80.

## FLOW DIAGRAM



NANDB(UN,UT,VR)

INPUT VARIABLE
VR elliptical angle defining point on cylinder ($\overline{XC}$) for which normal and tangent unit vectors are to be calculated

OUTPUT VARIABLES
UN x and y components of unit vector normal to cylinder at point $\overline{XC}$ in RCS
UT x and y components of unit vector tangent to cylinder at point $\overline{XC}$ in RCS

Calculate unit normal and tangent

Return

## SYMBOL DICTIONARY

UN X AND Y COMPONENTS OF UNIT VECTOR NORMAL TO CYLINDER IN RCS
UT X AND Y COMPONENTS OF UNIT VECTOR TANGENT TO CYLINDER IN RCS
VR ELL ANGLE IN ERCS DEFINING THE POINT ON CYLINDER FOR WHICH NORMAL AND TANGENT UNIT VECTORS ARE TO BE CALCULATED

# CODE LISTING

```
C-----------------------------------------------------------------------
      SUBROUTINE NANDB(UN,UT,VR)
C!!!
C!!!  COMPUTES NORMAL AND TANGENT VECTOR AT ANGLE VR ON THE
C!!!  ELLIPTIC CYLINDER.
C!!!
      DIMENSION UN(2),UT(2)
      COMMON/GEOMEL/A,B,ZC(2),SNC(2),CNC(2),CTC(2)
      DN=SQRT(A*A*SIN(VR)*SIN(VR)+B*B*COS(VR)*COS(VR))
      UN(1)=B*COS(VR)/DN
      UN(2)=A*SIN(VR)/DN
      UT(1)=-UN(2)
      UT(2)=UN(1)
      RETURN
      END
```

## OUTPUT

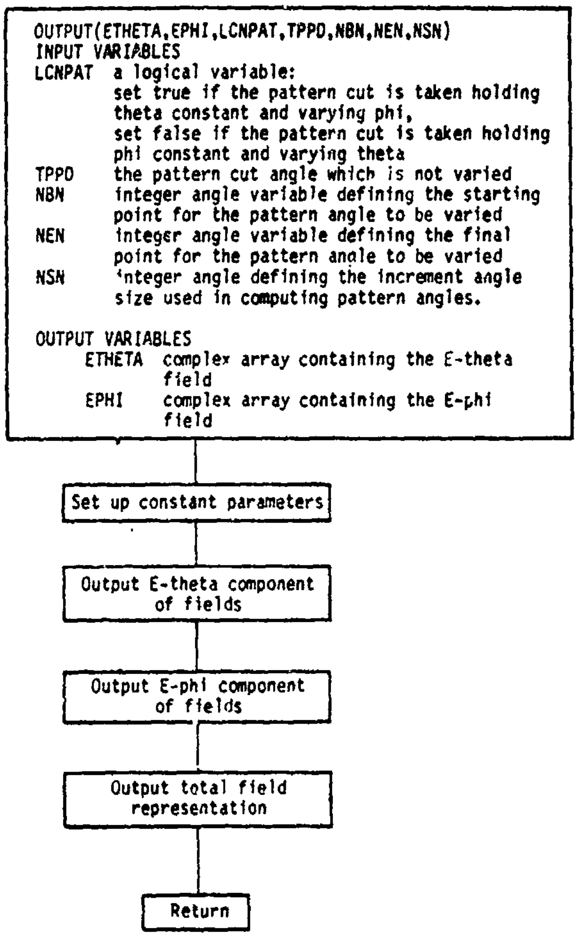
### PURPOSE

To output various representations of the computed fields on the line printer.

### METHOD

This subroutine outputs various representations of the fields on the line printer for a convenient analysis of the data calculated for a given pattern computation. The fields are represented in complex form, magnitude and phase, normalized and unnormalized, and in decibels. If the far field range is specified the fields are output in volts/meter. If no range is specified the fields are given in volts/unit. If the power radiated is specified the directive gain is given. If it is not specified the radiation intensity is output instead. Also, the major and minor components of the total fields are given, as well as the axial ratio and tilt angle of the polarization ellipse. Complete details of the output presentation are given in the User's Manual[8].

## FLOW DIAGRAM



```
OUTPUT(ETHETA,EPHI,LCNPAT,TPPD,NBN,NEN,NSN)
INPUT VARIABLES
LCNPAT  a logical variable:
        set true if the pattern cut is taken holding
        theta constant and varying phi,
        set false if the pattern cut is taken holding
        phi constant and varying theta
TPPD    the pattern cut angle which is not varied
NBN     integer angle variable defining the starting
        point for the pattern angle to be varied
NEN     integer angle variable defining the final
        point for the pattern anole to be varied
NSN     'integer angle defining the increment angle
        size used in computing pattern angles.

OUTPUT VARIABLES
     ETHETA  complex array containing the E-theta
             field
     EPHI    complex array containing the E-phi
             field
```

Set up constant parameters
Output E-theta component of fields
Output E-phi component of fields
Output total field representation
Return

## SYMBOL DICTIONARY

| | |
|---|---|
| AXRAT | AXIAL RATIO OF POLARIZATION ELLIPSE |
| EDIF2 | COMPUTATIONAL VARIABLE |
| EMAJ2 | MAJOR AXIS RADIATION INTENSITY *2*ZO |
| EMIN2 | MINOR AXIS RADIATION INTENSITY *2*ZO |
| EPHA | PHASE OF EPHI |
| EPHDB | E-PHI DIRECTIVE GAIN OR RADIATION INTENSITY |
| EPHDBN | NORMALIZED E-PHI GAIN OR INTENSITY |
| EPHI | COMPLEX ARRAY CONTAINING THE E-PHI FIELD |
| EPHM | MAGNITUDE OF EPHI |
| EPHMN | NORMALIZED E-PHI MAGNITUDE |
| EPHMR | MAGNITUDE OF EPHM WITH RANGE FACTOR |
| EPHMX | MAXIMUM MAGNITUDE OF EPHI |
| EPHPS | PHASE OF EPHR |
| EPHR | EPHI WITH RANGE FACTOR INCLUDED |
| ETDTMX | MAXIMUM MAGNITUDE OF THE RADIATION INTENSITY*2*ZO |
| ETHA | PHASE OF ETHETA |
| ETHDB | E-THETA DIRECTIVE GAIN OR RADIATION INTENSITY |
| ETHDBN | THETA NORMALIZED GAIN OR INTENSITY |
| ETHETA | COMPLEX ARRAY CONTAINING THE E-THETA FIELD |
| ETHM | MAGNITUDE OF ETHETA |
| ETHMN | NORMALIZED E-THETA MAGNITUDE |
| ETHMR | MAGNITUDE OF ETHM WITH RANGE FACTOR |
| ETHMX | MAXIMUM MAGNITUDE OF ETHETA |
| ETHPS | PHASE OF ETHR |
| ETHR | ETHETA WITH RANGE FACTOR |
| ETOT2 | RADIATION INTENSITY TIMES 2*ZO |
| ETOTN | NORMALIZED GAIN OR INTENSITY |
| FACP | GAIN OR INTENSITY FACTOR |
| FACPDB | FACP IN DB |
| FRANG | RANGE FACTOR |
| GLORBA | COMPUTATIONAL VARIABLE |
| GMAJ | MAJOR AXIS DIRECTIVE GAIN OR RADIATION INTENSITY IN DB |
| GMIN | MINOR AXIS DIRECTIVE GAIN OR RADIATION INTENSITY IN DB |
| GTOT | DIRECTIVE GAIN OR RADIATION INTENSITY IN DB |
| GTOTN | NORMALIZED GAIN OR INTENSITY IN DECIBELS |
| I | DO LOOP INDEX |
| IM | INTEGER VALUE OF ANGLE BEING VARIED |
| IMAX | NUMBER OF LINES TO BE OUTPUT BETWEEN SPACING |
| LCNPAT | LOGICAL VARIABLE RELATED TO THE PATTERN CUT TAKEN: LCNPAT=TRUE IF THETA IS FIXED AND PHI IS VARIED, AND LCNPAT=FALSE IF PHI IS FIXED AND THETA IS VARIED |
| NBM | ONE PLUS NBN |
| NBN | AN INTEGER DEFINING THE STARTING POINT OF PATTERN ANGLE TO BE VARIED |
| NEM | ONE PLUS NEN |
| NEN | AN INTEGER DEFINING THE ENDING POINT OF THE PATTERN ANGLE WHICH IS VARIED |
| NSN | AN INTEGER DEFINING THE INCREMENT IN THE PATTERN ANGLE WHICH IS VARIED BETWEEN STARTING AND END POINTS |
| PHI | FIXED PHI ANGLE |
| RANGL | RANGE PHASE VALUE |
| STILTA | SINE OF TILTA |
| THI | FIXED THETA ANGLE |
| TILTA | TILT ANGLE OF POLARIZATION ELLIPSE IN RADIANS |
| TILTD | TILT ANGLE IN DEGREES |
| TPPD | THE FIXED ANGLE DEFINING THE PATTERN CUT |

## CODE LISTING

```
C-------------------------------------------------------
      SUBROUTINE OUTPUT (ETHETA,EPHI.LCNPAT.TPPD,NBN,NEN,NSN)
C!!!
C!!!  THIS SUBROUTINE IS USED TO OUTPUT E-THETA AND E-PHI
C!!!  PATTERN DATA ON THE LINE PRINTER. IT IS OUTPUT WITH
C!!!  EACH PATTERN CALCULATION AS A PRINTED RECORD OF RESULTS.
C!!!
      COMPLEX ETHETA(1),EPHI(1),ETHR,EPHR,FRANG
      LOGICAL LPRAD,LRANG,LCNPAT
      COMMON/OUTPTP/LPRAD,LRANG,PRAD,RANG,ML
      COMMON/PIS/PI,TPI,DPR,RPD
100   FORMAT (1H ,'*************************************************
****
     2************************************************************
******
     2*'////)
101   FORMAT(3X,'*******      ******* *     * ******* *******    *
     2    ******     *    ******* ******* ******* ******  *     *'
)
102   FORMAT(3X,'*               *    *     * *          *      * *
     2    *     *   * *      *       *    *       *     * **    *'
)
103   FORMAT(3X,'*               *    *     * *          *     *   *
     2    *     *  *   *     *       *    *       *     * * *   *'
)
104   FORMAT(3X,'******  ----    *    ******* ******     *    *
*
     2    ******  *     *    *       *    ******  ******  *  *  *'
)
105   FORMAT(3X,'*               *    *     * *          *    ******
*
     2    *       *******    *       *    *       *   *   *   *  *'
)
106   FORMAT(3X,'*               *    *     * *          *    *
*
     2    *       *     *    *       *    *       *    *  *    **'
)
107   FORMAT(3X,'*******         *    *     * *******    *    *
*
     2    *       *     *    *       *    ******* *     * *     *'
///)
201   FORMAT (3X,'*******      ******  *     *  *****
     2     ******     *    ******* ******* ******* ******  *     *
')
202   FORMAT(3X,'*            *     * *     *    *
     2    *     *   * *      *       *    *       *     * **    *'
)
203   FORMAT(3X,'*            *     * *     *    *
     2    *     *  *   *     *       *    *       *     * * *   *'
)
204   FORMAT(3X,'******  ---- ******  *******    *
     2    ******  *     *    *       *    ******  ******  *  *  *'
)
205   FORMAT(3X,'*            *       *     *    *
     2    *       *******    *       *    *       *   *   *   *  *'
)
206   FORMAT(3X,'*            *       *     *    *
     2    *       *     *    *       *    *       *    *  *    **'
)
207   FORMAT (3X,'*******      *       *     *  *****
     2     *       *     *    *       *    ******* *     * *     *
'
     2////)
150   FORMAT(' THE FIELDS ARE REFERENCED TO THE PATTERN COORDINATE',
     2' SYSTEM',///)
151   FORMAT (1H ,12X,'UNNORMALIZED',15X,'NORMALIZED')
```

```
 152   FORMAT (6X,'THETA',9X,'PHI',16X,'E-PHI',14X,'PHASE',7X,
      2'MAGNITUDE',4X,'DB GAIN',6X,'MAGNITUDE',7X,'DB')
 153   FORMAT (6X,'THETA',9X,'PHI',15X,'E-THETA',13X,'PHASE',7X,
      2'MAGNITUDE',4X,'DB GAIN',6X,'MAGNITUDE',7X,'DB')
 155   FORMAT (6X,'THETA',9X,'PHI',15X,'E-THETA',13X,'PHASE',7X,
      2'MAGNITUDE',3X,'DB INTEN.',5X,'MAGNITUDE',7X,'DB')
 156   FORMAT(6X,'THETA',9X,'PHI',16X,'E-PHI',14X,'PHASE',7X,
      2'MAGNITUDE',3X,'DB INTEN.',5X,'MAGNITUDE',7X,'DB')
 154   FORMAT (2(3X,'----------'),3X,'------------'
      2,'----------',3X,'----------',2(3X,'------------',
      2'----------')/)
 300   FORMAT (1H1)
 400   FORMAT (1H0)
 500   FORMAT (3X,9(F10.5,3X))
 501   FORMAT(2(3X,F10.5),2(2X,E11.5),3X,F10.5,2X,E11.5,3(3X,F10.5))
C!!!   SET UP CONSTANTS
       NBM=NBN+1
       NEM=NEN+1
       IF(LCNPAT) THI=TPPD
       FRANG=CMPLX(1.,0.)
       IF(.NOT.LRANG) GO TO 600
       RANGL=RANG/WL-AINT(RANG/WL)
       FRANG=CEXP(CMPLX(0.,-TPI*RANGL))/RANG
 600   CONTINUE
       FACP=1./(240.*PI)
       IF(LPRAD) FACP=1./(60.*PRAD)
       FACPDB=10.*BLOG10(FACP)
       ETHMX = EABS(ETHETA(1))
       EPHMX = EABS(EPHI(1))
       ETOTMX=ETHMX*ETHMX+EPHMX*EPHMX
       DO 1 I = NBM,NEM,NSN
       ETHM = EABS(ETHETA(I))
       IF (ETHM .GT. ETHMX)  ETHMX = ETHM
       EPHM = EABS(EPHI(I))
       IF (EPHM .GT. EPHMX) EPHMX = EPHM
       ETOT2=ETHM*ETHM+EPHM*EPHM
       IF(ETOT2.GT.ETOTMX) ETOTMX=ETOT2
 1     CONTINUE
C!!!   OUTPUT E-THETA REPRESENTATIONS
       WRITE(6,300)
       WRITE(6,100)
       WRITE (6,100)
       WRITE (6,101)
       WRITE (6,102)
       WRITE (6,103)
       WRITE (6,104)
       WRITE (6,105)
       WRITE (6,106)
       WRITE (6,107)
       WRITE (6,150)
       WRITE(6,151)
       IF(LPRAD) WRITE (6,153)
       IF(.NOT.LPRAD) WRITE(6,155)
       WRITE (6,154)
       IMAX=10*NSN+1
       DO 2 I = NBM,NEM,NSN
       IM=I-1
       IF(LCNPAT) PHI=IM
       IF(LCNPAT) GO TO 25
       IF(IM.GT.180) GO TO 24
       PHI=TPPD
       THI=IM
       GO TO 25
 24    PHI=TPPD+180.
       IF(PHI.GE.360.) PHI=PHI-360.
       THI=360-IM
```

```
   25 CONTINUE
      ETHR=ETHETA(I)*FRANG
      ETHM = BABS(ETHETA(I))
      ETHMR=ETHM/RANG
      ETHPS = DPR*BTAN2(AIMAG(ETHR),REAL(ETHR))
      ETHDB = 2Ø.*BLOG1Ø(ETHM)+FACPDB
      ETHMN = ETHM/ETHMX
      ETHDBN = 2Ø.*BLOG1Ø(ETHMN)
      IF (I .GT. IMAX)  IMAX = IMAX+1Ø*NSN
      WRITE (6,5Ø1) THI,PHI,ETHR,ETHPS,ETHMR,ETHDB,ETHMN,ETHDBN
      IF (I .EQ. IMAX)  WRITE (6,4ØØ)
    2 CONTINUE
C!!!  OUTPUT E-PHI REPRESENTATIONS
      WRITE(6,1ØØ)
      WRITE (6,1ØØ)
      WRITE (6,3ØØ)
      WRITE(6,1ØØ)
      WRITE (6,1ØØ)
      WRITE (6,2Ø1)
      WRITE (6,2Ø2)
      WRITE (6,2Ø3)
      WRITE (6,2Ø4)
      WRITE (6,2Ø5)
      WRITE (6,2Ø6)
      WRITE (6,2Ø7)
      WRITE (6,15Ø)
      WRITE (6,151)
      IF(LPRAD) WRITE (6,152)
      IF(.NOT.LPRAD) WRITE(6,156)
      WRITE (6,154)
      IMAX=1Ø*NSN+1
      DO 3 I = NBM,NEM,NSN
      IM=I-1
      IF(LCNPAT) PHI=IM
      IF(LCNPAT) GO TO 35
      IF(IM.GT.18Ø) GO TO 34
      PHI=TPPD
      THI=IM
      GO TO 35
   34 PHI=TPPD+18Ø.
      IF(PHI.GE.36Ø.) PHI=PHI-36Ø.
      THI=36Ø-IM
   35 CONTINUE
      EPHR=EPHI(I)*FRANG
      EPHM = BABS(EPHI(I))
      EPHMR=EPHM/RANG
      EPHPS = DPR*BTAN2(AIMAG(EPHR),REAL(EPHR))
      EPHDB = 2Ø.*BLOG1Ø(EPHM)+FACPDB
      EPHMN = EPHM/EPHMX
      EPHDBN = 2Ø.*BLOG1Ø(EPHMN)
      IF (I .GT. IMAX)  IMAX = IMAX+1Ø*NSN
      WRITE (6,5Ø1) THI,PHI,EPHR,EPHPS,EPHMR,EPHDB,EPHMN,EPHDBN
      IF (I .EQ. IMAX)  WRITE (6,4ØØ)
    3 CONTINUE
C!!!  OUTPUT TOTAL FIELD REPRESENTATIONS
      WRITE (6,1ØØ)
      WRITE(6,1ØØ)
      WRITE(6,3ØØ)
      WRITE(6,1ØØ)
      WRITE(6,1ØØ)
      IF(LPRAD) WRITE(6,3Ø1)
  3Ø1 FORMAT(' TOTAL DIRECTIVE GAIN IN DB '///)
      IF(.NOT.LPRAD) WRITE(6,3Ø3)
  3Ø3 FORMAT(' TOTAL RADIATION INTENSITY IN DB.'///)
      WRITE(6,15Ø)
      IF(LPRAD) WRITE(6,3Ø2)
  302 FORMAT(6X,'THETA',9X,'PHI',9X,'MAJOR',8X,'MINOR',7X
     2,'TILT ANG',4X,'AXIAL RATIO',2X,'TOTAL GAIN',4X,'NORM GAIN')
      IF(.NOT.LPRAD) WRITE(6,304)
  304 FORMAT(6X,'THETA',9X,'PHI',9X,'MAJOR',8X,'MINOR',7X
     2,'TILT ANG',4X,'AXIAL RATIO',2X,'TOTAL INTEN.',2X,'NORM',
     2' INTEN.')
      IMAX=10*NSN+1
      DO 4 I=NBM,NEM,NSN
      IM=I-1
      IF(LCNPAT) PHI=IM
      IF(LCNPAT) GO TO 45
      IF(IM.GT.18Ø) GO TO 44
      PHI=TPPD
      THI=IM
      GO TO 45
   44 PHI=TPPD+18Ø.
      IF(PHI.GE.36Ø.) PHI=PHI-36Ø.
      THI=36Ø-IM
   45 CONTINUE
      ETHM=BAES(ETHETA(I))
      EPHM=BAES(EPHI(I))
      ETOT2=ETHM*ETHM+EPHM*EPHM
      GTOT=1Ø.*BLOG1Ø(FACP*ETOT2)
      ETOTN=ETOT2/ETOTMX
      GTOTN=1Ø.*BLOG1Ø(ETOTN)
      IF(I.GT.IMAX) IMAX=IMAX+1Ø*NSN
      EPHA=BTAN2(AIMAG(EPHI(I)),REAL(EPHI(I)))
      ETHA=BTAN2(AIMAG(ETHETA(I)),REAL(ETHETA(I)))
      GLURBA=2.*EPHM*ETHM*COS(EPHA-ETHA)
      EDIF2=ETHM*ETHM-EPHM*EPHM
      TILTA=.5*BTAN2(GLURBA,EDIF2)
      TILTD=DPR*TILTA
      STILTA=SIN(TILTA)
      EMAJ2=-EDIF2*STILTA*STILTA+GLURBA*STILTA*COS(TILTA)+
     2ETHM*ETHM
      GMAJ=1Ø.*BLOG1Ø(EMAJ2)+FACPDB
      EMIN2=EDIF2*STILTA*STILTA-GLURBA*STILTA*COS(TILTA)+
     2EPHM*EPHM
      GMIN=1Ø.*BLOG1Ø(EMIN2)+FACPDB
      AXRAT=SQRT(ABS(EMIN2/EMAJ2))
      WRITE(6,5ØØ) THI,PHI,GMAJ,GMIN,TILTD,AXRAT,GTOT,GTOTN
      IF(I.EQ.IMAX) WRITE(6,4ØØ)
    4 CONTINUE
      WRITE(6,1ØØ)
      WRITE(6,1ØØ)
      RETURN
      END
```

## PATROT

PURPOSE

To convert pattern angles from pattern cut coordinate system to reference coordinate system representation.

PERTINENT GEOMETRY

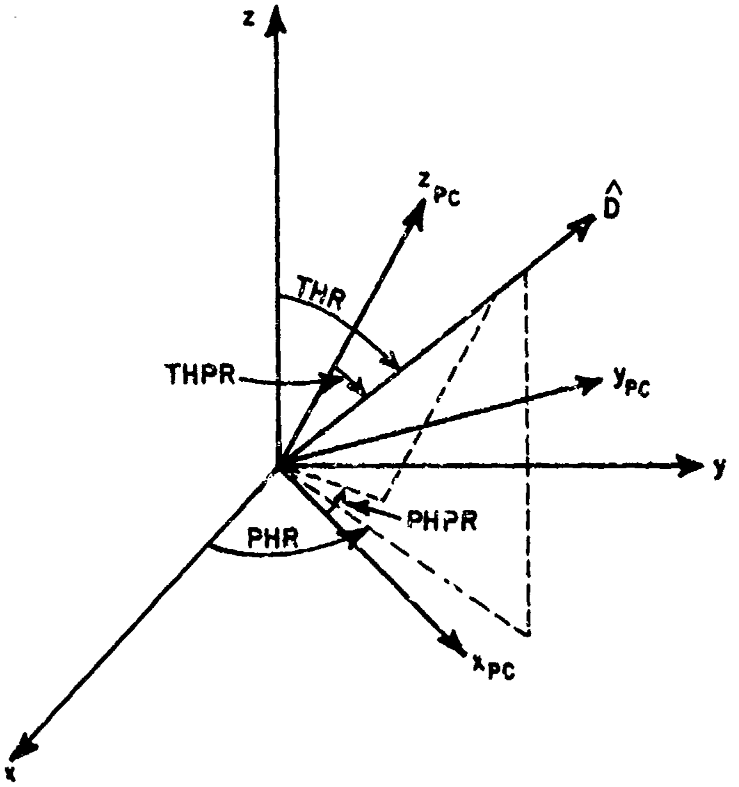


Figure 81--Illustration of propagation direction $\hat{D}$ and reference and pattern-cut coordinate systems.

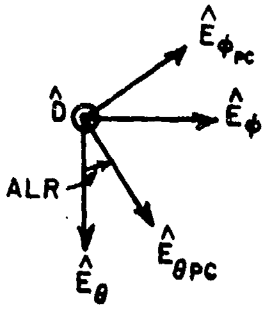


Figure 82--Illustration of polarization rotation angle ALR.

METHOD

The observation direction is defined in the pattern cut coordinate system as

$$\hat{D} = \cos(PHPR)\sin(THPR)\hat{x}_p + \sin(PHPR)\sin(THPR)\hat{y}_p + \cos(THPR)\hat{z}_p.$$

This is converted into the reference coordinate system as

$$\hat{D} = (\hat{D}\cdot\hat{x})\hat{x} + (\hat{D}\cdot\hat{y})\hat{y} + (\hat{D}\cdot\hat{z})\hat{z}$$

or

$$\hat{D} = \cos(PHR)\sin(THR)\hat{x} + \sin(PHR)\sin(THR)\hat{y} + \cos(THR)\hat{z}.$$

The polarization conversion angle is given by

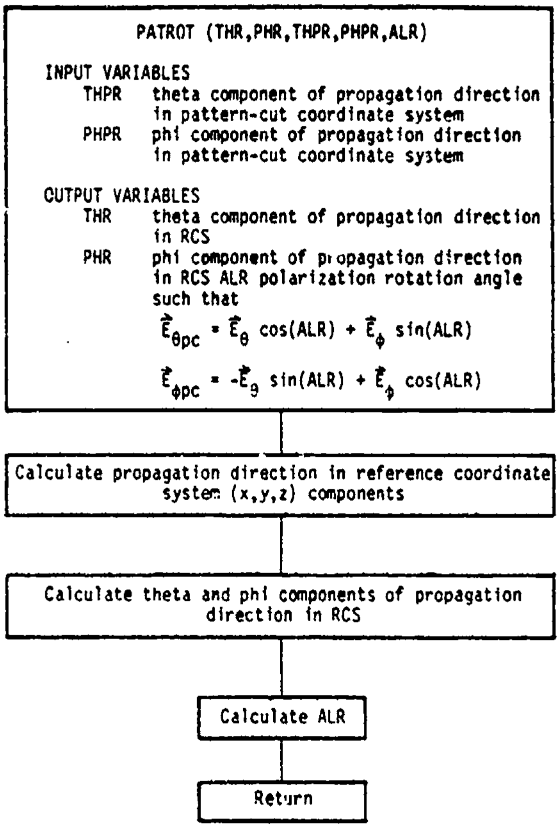
$$ALR = \tan^{-1} \frac{\hat{\theta}_{pc}\cdot\hat{\phi}}{\hat{\theta}_{pc}\cdot\hat{\theta}}$$

so that after the scattered fields are computed they can be converted back to the pattern cut coordinate system using

$$\vec{E}_{\theta pc} = \vec{E}_\theta \cos(ALR) + \vec{E}_\phi \sin(ALR)$$

$$\vec{E}_{\phi pc} = -\vec{E}_\theta \sin(ALR) + \vec{E}_\phi \cos(ALR)\ .$$

## FLOW DIAGRAM



PATROT (THR,PHR,THPR,PHPR,ALR)

INPUT VARIABLES

THPR theta component of propagation direction in pattern-cut coordinate system

PHPR phi component of propagation direction in pattern-cut coordinate system

OUTPUT VARIABLES

THR theta component of propagation direction in RCS

PHR phi component of propagation direction in RCS ALR polarization rotation angle such that

$$\vec{E}_{\theta pc} = \vec{E}_{\theta} \cos(ALR) + \vec{E}_{\phi} \sin(ALR)$$

$$\vec{E}_{\phi pc} = -\vec{E}_{\theta} \sin(ALR) + \vec{E}_{\phi} \cos(ALR)$$

Calculate propagation direction in reference coordinate system (x,y,z) components

Calculate theta and phi components of propagation direction in RCS

Calculate ALR

Return

## SYMBOL DICTIONARY

| | |
|---|---|
| ALR | POLARIZATION ROTATION ANGLE |
| CPH | COS(PHR) |
| CPHP | COS(PHPR) |
| CTH | COS(THR) |
| CTHP | COS(THPR) |
| PDTP | COMPUTATIONAL VARIABLE |
| PHPR | PHI COMPONENT OF PROPAGATION DIRECTION IN PATTERN CUT COORDINATE SYSTEM |
| PHR | PHI COMPONENT OF PROPAGATION DIRECTION IN RCS |
| RDX<br>RDY<br>RDZ | X,Y,AND Z COMPONENTS OF PROPAGATION DIRECTION IN RCS |
| SPH | SIN(PHR) |
| SPHP | SIN(PHPR) |
| STH | SIN(THR) |
| STHP | SIN(THPR) |
| TDTP | COMPUTATIONAL VARIABLE |
| THPR | THETA COMPONENT OF PROPAGATION DIRECTION IN PATTERN CUT COORD SYSTEM |
| THR | THETA COMPONENT OF PROPAGATION DIRECTION IN RCS |
| TX<br>TY<br>TZ | X,Y,Z COMPONENTS OF THETA POLARIZATION UNIT VECTOR OF PATTERN CUT COORDINATE SYSTEM IN RCS COMPONENTS |

# CODE LISTING

```
C-----------------------------------------------------------------------
      SUBROUTINE PATROT(THR,PHR,THPR,PHPR,ALR)
C!!!
C!!!
C!!!  ROTATION OF PATTERN ANGLES FROM PATTERN AXES (THP,PHP)
C!!!  TO REFERENCE AXES (TH,PH). NOTE THAT ALR IS DEFINED BY:
C!!!    E-THETAP=E-THETA*COS(ALR)+E-PHI*SIN(ALR)
C!!!    E-PHIP=-E-THETA*SIN(ALR)+E-PHI*COS(ALR)
C!!!
C!!!
      LOGICAL LDEBUG,LTEST
      COMMON/TEST/LDEBUG,LTEST
      COMMON/PIS/PI,TPI,DPR,RPD
      COMMON/PATDAT/XPC(3),YPC(3),ZPC(3)
      STHP=SIN(THPR)
      CTHP=COS(THPR)
      SPHP=SIN(PHPR)
      CPHP=COS(PHPR)
C!!!  CALCULATE PROPAGATION DIRECTION IN REFERENCE COORDINATE
C!!!  SYSTEM (X,Y,Z) COORDINATES
      RDX=STHP*CPHP*XPC(1)+STHP*SPHP*YPC(1)+CTHP*ZPC(1)
      RDY=STHP*CPHP*XPC(2)+STHP*SPHP*YPC(2)+CTHP*ZPC(2)
      RDZ=STHP*CPHP*XPC(3)+STHP*SPHP*YPC(3)+CTHP*ZPC(3)
      SQN=SQRT(RDX*RDX+RDY*RDY)
C!!!  CALCULATE THR AND PHR
      THR=BTAN2(SQN,RDZ)
      PHR=BTAN2(RDY,RDX)
      STH=SIN(THR)
      CTH=COS(THR)
      SPH=SIN(PHR)
      CPH=COS(PHR)
      TX=CTHP*CPHP*XPC(1)+CTHP*SPHP*YPC(1)-STHP*ZPC(1)
      TY=CTHP*CPHP*XPC(2)+CTHP*SPHP*YPC(2)-STHP*ZPC(2)
      TZ=CTHP*CPHP*XPC(3)+CTHP*SPHP*YPC(3)-STHP*ZPC(3)
C!!!  CALCULATE ALR
      TDTP=TX*CTH*CPH+TY*CTH*SPH-TZ*STH
      PDTP=-TX*SPH+TY*CPH
      ALR=BTAN2(PDTP,TDTP)
      IF (.NOT.LTEST) GO TO 1
      WRITE(6,2)
 2    FORMAT(/,' TESTING PATROT SUBROUTINE')
      WRITE (6,*) THR,PHR,THPR,PHPR,ALR
  1   RETURN
      END
```

## PFUN

PURPOSE

This function computes the p* function for the cylinder's acoustically soft diffraction coefficient.

METHOD

The p* function is defined as [14,15]

$$p^*(x) = \frac{1}{2\sqrt{\pi x}} + \hat{P}_s(x)\, e^{j\ \pi/4}$$

where

$$\hat{P}_s(x) = \frac{e^{-j\ \pi/4}}{\sqrt{\pi}} \int_{-\infty}^{\infty} \frac{V(\tau)}{w_2(\tau)}\, e^{-jx\tau}\, d\tau \quad ,$$

and $V(\tau)$ and $w_2(\tau)$ are Fock type Airy functions. The p* function is computed as follows:

1) for $x \leq -3$

$$p^*(x) = \frac{1}{2\sqrt{\pi x}} + \frac{1}{2}\sqrt{|x|}\left(1+j\,\frac{2}{x^3}\right) e^{j\frac{x^3}{12}}\, e^{j\ \pi/4}$$

2) for $-3 < x < 2$

$$p^*(x) = p^*(x_i) + \frac{(x-x_i)}{(x_{i+1}-x_i)}\,(p^*(x_{i+1})-p^*(x_i)),$$

where the $p^*(x_i)$ are tabulated values[14,15] and $x_{i+1}-x_i=0.1$ with $x_i \leq x \leq x_{i+1}$.

3) For $x \geq 2$

$$p^*(x) = \frac{1}{2\sqrt{\pi x}} - \frac{e^{j\ \pi/6}}{2\sqrt{\pi}} \sum_{n=1}^{5} \frac{\epsilon^{x q_n e^{-j\frac{5\pi}{6}}}}{[A_i'(-q_n)]^2}$$

where $A_i'(\tau)$ is the derivative of the Miller type Airy function.

## SYMBOL DICTIONARY

| | |
|---|---|
| AMC | -0.5*CEXP(J*PI/6)/SQRT(PI) |
| AQ | DERIVATIVE OF MILLER TYPE AIRY FUNCTION AT Q |
| C | 0.5/SQRT(PI) |
| EXC | CEXP(-5*PI/6) |
| I | SMALLEST INTEGER CLOSEST TO 10*X |
| Q | ZEROES OF MILLER TYPE AIRY FUNCTION |
| PFUN | P FUNCTION |
| PJ | IMAGINARY PART OF TABULATED P FUNCTION |
| PR | REAL PART OF TABULATED P FUNCTION |
| X | ARGUMENT OF P FUNCTION |
| XI | REAL NUMBER REPRESENTATION OF I |

## CODE LISTING

```
C-------------------------------------------------------------------
      COMPLEX FUNCTION PFUN(X)
      DIMENSION PR(51),PJ(51)
C!!!
C!!!  COMPUTES THE P FUNCTION OF THE CYLINDER'S
C!!!  DIFFRACTION COEFFICIENT (SOFT CASE)
C!!!
      COMPLEX AMC,EXC
      DIMENSION Q(5),AQ(5)
      COMMON/PIS/PI,TPI,DPR,RPD
      DATA AMC,EXC/(-0.24430,-0.14105),(-0.866025,-0.5)/
      DATA C/0.28209/
      DATA Q/2.33811,4.08795,5.52056,6.78671,7.94413/
      DATA AQ/0.70121,-0.80311,0.86520,-0.91085,0.94734/
      DATA PR/-.054,.125,.276,.399,.4[illegible],.560,.605,.629,.638,.636
     2,.624,.606,.584,.560,.536,.510,.487,.464,.444,.425,.408
     2,.393,.379,.367,.357,.347,.338,.330,.322,.314,.307,.299
     2,.292,.284,.276,.268,.259,.251,.242,.234,.224,.215,.206
     2,.198,.190,.180,.173,.165,.158,.150,.144/
      DATA PJ/.879,.840,.769,.678,.577,.469,.354,.265,.173,.091
     2,.019,-.043,-.113,-.139,-.174,-.202,-.224,-.240,-.251
     2,-.257,-.260,-.260,-.256,-.252,-.244,-.236,-.225,-.214
     2,-.202,-.190,-.177,-.164,-.151,-.138,-.125,-.113,-.101
     2,-.090,-.080,-.070,-.061,-.053,-.045,-.039,-.032,-.027
     2,-.023,-.018,-.014,-.011,-.010/
      IF(X.LE.-3.)GO TO 1
      IF(X.GE.2.)GO TO 2
      I=((3.+X)*10.)
      XI=FLOAT(I)-30.
      I=I+1
      PFUN=CMPLX(PR(I),-PJ(I))+(10.*X-XI)*CMPLX(PR(I+1)-PR(I),
     2-PJ(I+1)+PJ(I))
      RETURN
    1 PFUN=.5*(1./(SQRT(PI)*X)+SQRT(ABS(X))*CEXP(CMPLX(0.,0.25
     2*PI+X*X*X/12.))*CMPLX(1.,2./(X*X*X)))
      RETURN
    2 PFUN=(0.,0.)
      DO 3 N=1,5
    3 PFUN=PFUN+CEXP(X*Q(N)*EXC)/AQ(N)/AQ(N)
      PFUN=PFUN*AMC+C/X
      RETURN
      END
```

## PLAINT

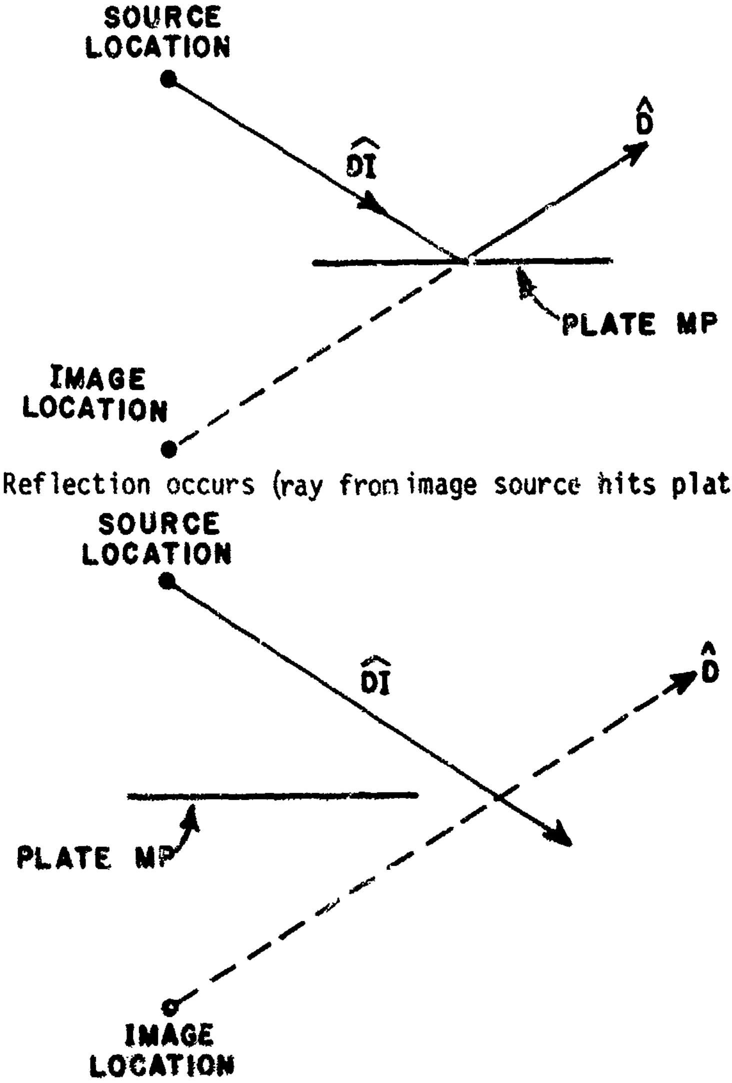
### PURPOSE

To determine if a ray traveling from a given source location in a given direction will intersect a given plate (or set of plates).

Note: several modes of operation are available:

```
If MH=-MP then only plate MP is checked (MP>0)
IF MH=0   all plates are checked
If MH=MP  all plates except plate MP are checked.
```

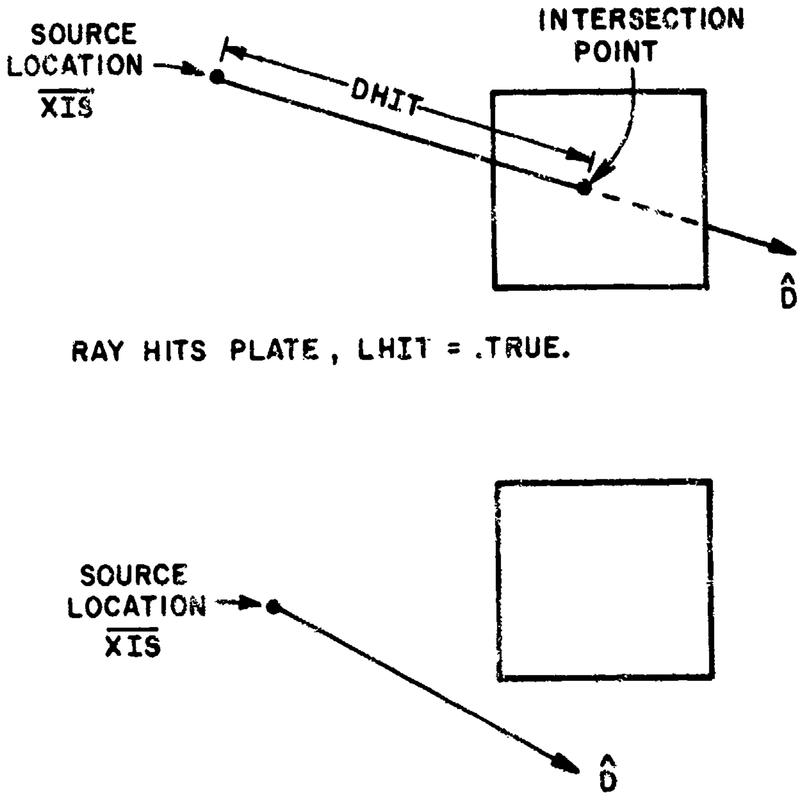
PERTINENT GEOMETRY



Reflection occurs (ray fron image source hits plate MP)

Reflection does not occur (ray from image source does not hit plate MP)

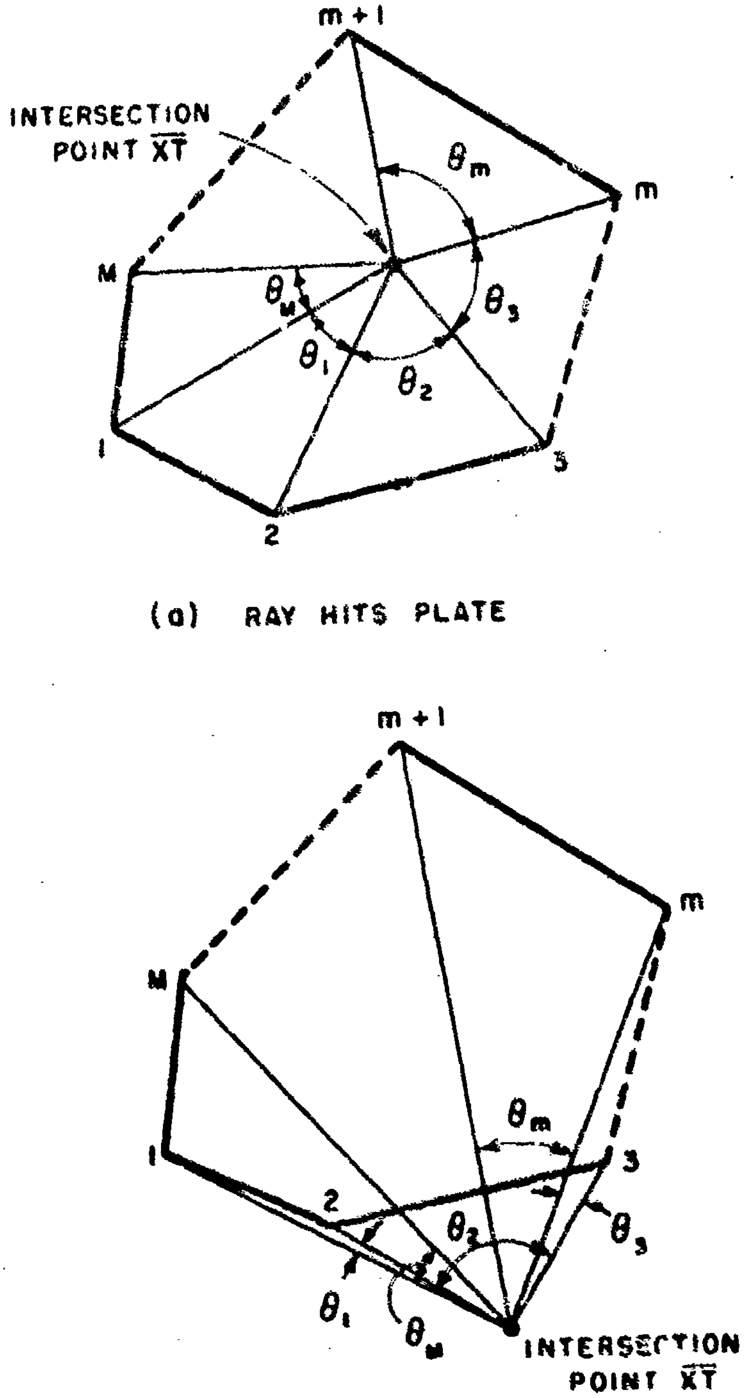
Figure 93--Geometry for determining if reflection from a given plate occurs.


SOURCE LOCATION $\overline{XIS}$
DHIT
INTERSECTION POINT
$\hat{D}$
SOURCE LOCATION $\overline{XIS}$
$\hat{D}$

RAY HITS PLATE, LHIT = .TRUE.

RAY DOES NOT HIT PLATE, LHIT = .FALSE.

Figure 84--Geometry for determining if a ray does or does not hit plate.



(a) RAY HITS PLATE

(b) RAY DOES NOT HIT PLATE

Figure 85--Geometry for deciding whether ray which hits plate plane hits finite plate.

## METHOD

This subroutine is used for a number of functions:

1. To determine if a source ray reflection from plate MP occurs. If a ray traveling from the source image location in the reflected ray direction passes thorugh plate MP, the reflection will occur (see Figure 83). The routine only checks plate MP (set MH=-MP). Note that the hit point (which is returned through the subroutine window) is the reflection point, and is used in shadowing tests.

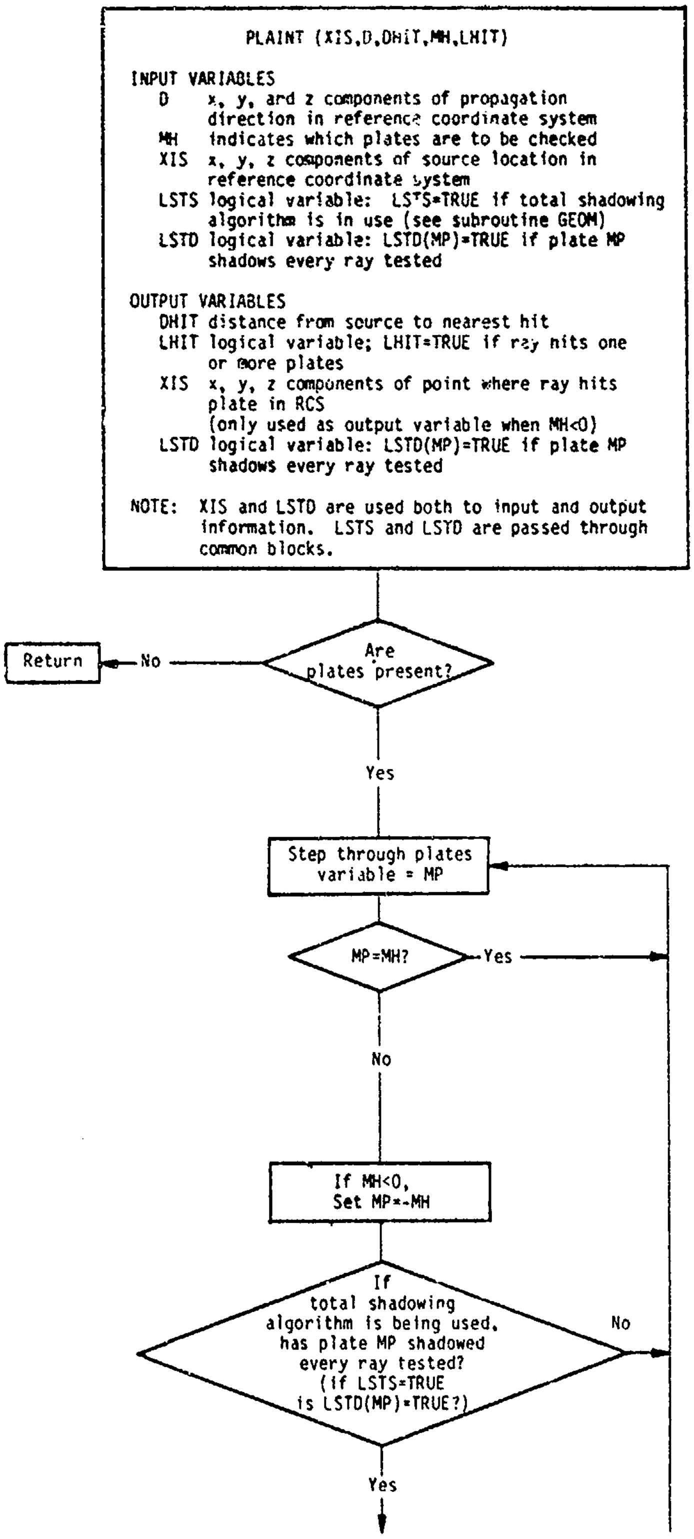
2. To test to see if a ray is shadowed between scatter points (or between the source and a scatter point). The routine checks all plates (set MH=0) and records the distance from the first scatter (or source) position to the nearest hit (if the ray hits any of the plates). If the distance to the nearest hit is shorter than the distance between scatter points (or between the source and scatter point), the ray is shadowed, and the GTD term being computed is set to zero. Otherwise, the ray is not disturbed and computations are carried out. Note that if the first scatter point is a reflection or diffraction point on a plate, all plates except that plate are checked (set MH=MP).

3. To determine if ray after final scatter point (or source ray) is shadowed. If the final scatter point is a cylinder (or if the source field is being computed) all plates are checked. If the final scatter point is on plate MP, all plates except plate MP are checked. If the ray hits a plate (LHIT=TRUE) the ray is shadowed and the GTD term is set to zero. If LHIT=FALSE, the ray is not shadowed and propagates undisturbed.

4. To determine if any one plate totally shadows plate MP from the source (referred to as the "total shadowing algorithm"). The routine checks all plates except plate MP (set MH=MP) and remembers plates which shadow the ray every time the routine is called (see section 6 of subroutine GEOM). The total shadowing algorithm is activated when LSTS is set TRUE.

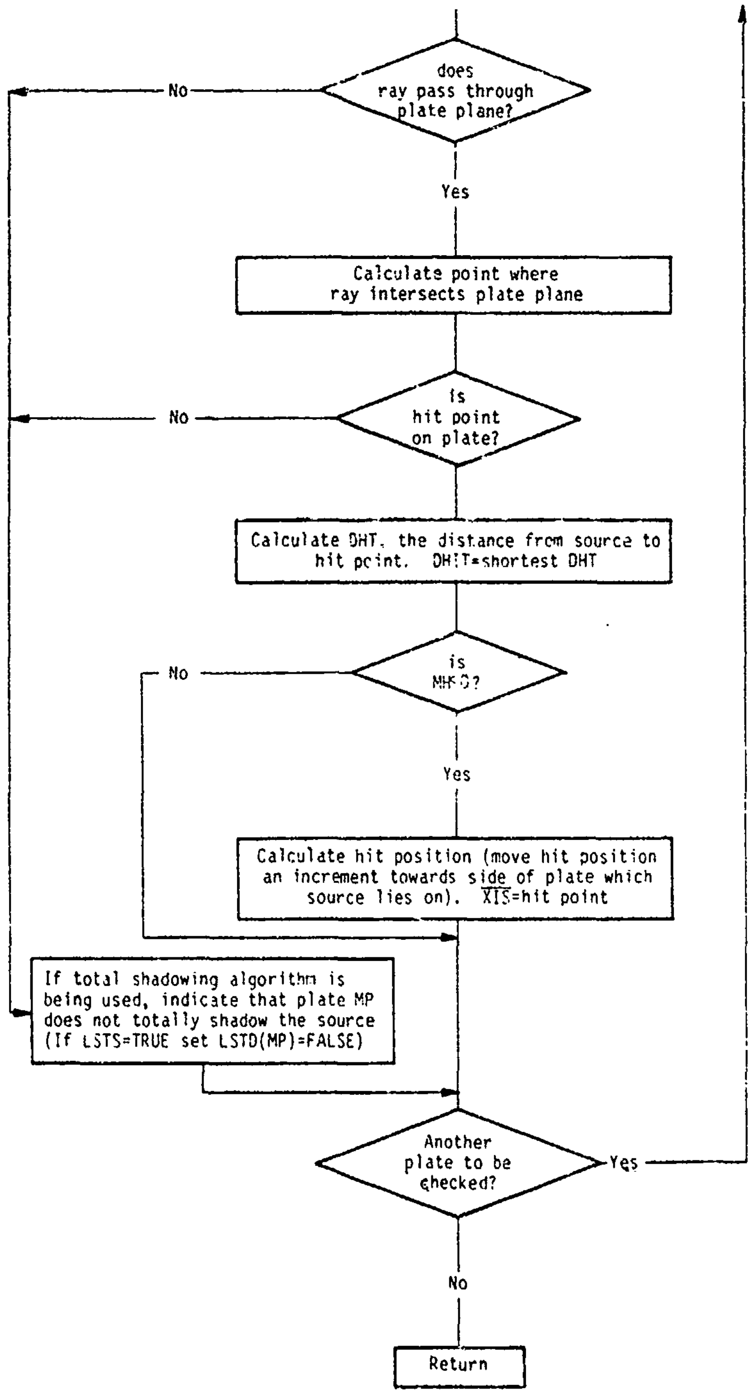
The hit algorithm first tests to see if a ray in the scatter direction will intersect the plane which the plate lies in by comparing the signs of the dot product of the scatter direction and the plate normal and the dot product of the vector from the source to a corner of the plate and the plate normal. If a hit is possible the intersection point on the plate plane is determined. Whether the intersection point lies within the bounds of the plate is tested by summing the angles formed by the vectors from the intersection point to the various corners of the plate as shown in Figure 85. If the sum is zero the intersection point does not fall within the bounds of the plate. If the sum is $2\pi$, the intersection point does fall within the bounds and the ray hits the plate. (See pp. 38-41, Reference 1).

## FLOW DIAGRAM



```
                PLAINT (XIS,D,DHIT,MH,LHIT)

INPUT VARIABLES
   D     x, y, and z components of propagation
         direction in reference coordinate system
   MH    indicates which plates are to be checked
   XIS   x, y, z components of source location in
         reference coordinate system
   LSTS  logical variable:  LSTS=TRUE if total shadowing
         algorithm is in use (see subroutine GEOM)
   LSTD  logical variable: LSTD(MP)=TRUE if plate MP
         shadows every ray tested

OUTPUT VARIABLES
   DHIT  distance from source to nearest hit
   LHIT  logical variable; LHIT=TRUE if ray nits one
         or more plates
   XIS   x, y, z components of point where ray hits
         plate in RCS
         (only used as output variable when MH<0)
   LSTD  logical variable: LSTD(MP)=TRUE if plate MP
         shadows every ray tested

NOTE:  XIS and LSTD are used both to input and output
       information.  LSTS and LSYD are passed through
       common blocks.
```

Return
No
Are plates present?
Yes
Step through plates variable = MP
MP=MH?
Yes
No
If MH<0, Set MP=-MH
If total shadowing algorithm is being used, has plate MP shadowed every ray tested? (if LSTS=TRUE is LSTD(MP)=TRUE?)
No
Yes


does
ray pass through
plate plane?
No
Yes
Calculate point where
ray intersects plate plane
is
hit point
on plate?
No
Calculate DHT, the distance from source to
hit point. DHIT=shortest DHT
is
MH=0?
No
Yes
Calculate hit position (move hit position
an increment towards side of plate which
source lies on). XIS=hit point
If total shadowing algorithm is
being used, indicate that plate MP
does not totally shadow the source
(If LSTS=TRUE set LSTD(MP)=FALSE)
Another
plate to be
checked?
Yes
No
Return

## SYMBOL DICTIONARY

| | |
|---|---|
| AN | DOT PRODUCT OF VECTOR FROM EDGE 1 OF PLATE MP TO SOURCE AND PLATE UNIT NORMAL |
| CP | COMPUTATIONAL VARIABLE |
| D | X,Y, AND Z COMPONENTS OF PROPAGATION DIRECTION IN REFERENCE COORDINATE SYSTEM |
| DBI | COMPUTATIONAL VARIABLE |
| DBT | COMPUTATIONAL VARIABLE |
| DHIT | DISTANCE FROM SOURCE TO NEAREST HIT |
| DHT | DISTANCE FROM SOURCE TO HIT POINT |
| DN | DOT PRODUCT OF PROPAGATION DIRECTION UNIT VECTOR AND PLATE UNIT NORMAL |
| LHIT | LOGICAL VARIABLE (SET TRUE IF RAY HITS AT LEAST ONE PLATE) |
| LSTD | SET TRUE IF PLATE MP TOTALLY SHADOWS PLATE MH FROM THE SOURCE |
| LSTS | SET TRUE IF TOTAL SHADOWING ROUTINE IS BEING USED |
| ME | DO LOOP VARIABLE |
| MEX | NUMBER OF EDGES ON PLATE MP |
| MH | SHOWS WHICH PLATES ARE TO BE CHECKED:<br>MH=-MP ONLY PLATE MP IS CHECKED<br>MH=Ø ALL PLATES ARE CHECKED<br>MH=MP ALL PLATES EXCEPT MP ARE CHECKED |
| MP | INDEX VARIABLE (NUMBER OF PLATE BEING CHECKED) |
| MPH | INDEX VARIABLE |
| MPP | DO LOOP VARIABLE |
| N | DO LOOP VARIABLE |
| RD | COMPUTATIONAL VARIABLE |
| XIS | X,Y,Z COMPONENTS OF SOURCE LOCATION IN REFERENCE COORDINATE SYSTEM (ENTERING ROUTINE)<br>X,Y,Z COMPONENTS OF HIT POSITION (LEAVING ROUTINE) |
| XT | X,Y,Z COMPONENTS OF POINT WHERE RAY INTERSECTS PLATE PLANE |

## CODE LISTING

```
C----------------------------------------------------------------------
      SUBROUTINE PLAINT(XIS,D,DHIT,MH,LHIT)
C!!!
C!!!  DOES RAY HIT PLATE.IF MH=0 ALL PLATES ARE CHECKED.
C!!!  IF MH=-MP THEN ONLY MP CHECKED AND SOURCE POSITION
C!!!  MOVED TO HIT POSITION IF RAY HITS MP.
C!!!  IF MH=MP,THEN ALL PLATES OTHER THAN MP ARE CHECKED.
C!!!
      DIMENSION XIS(3),D(3),XT(3)
      LOGICAL LHIT,LPLA,LCYL,LSTS,LSTD
      LOGICAL LGRND,LDEBUG,LTEST
      COMMON/TEST/LDEBUG,LTEST
      COMMON/GEOPLA/X(14,6,3),V(14,6,3),VP(14,6,3),VN(14,3)
     2,MEP(14),MPX
      COMMON/PIS/PI,TPI,DPR,RPD
      COMMON/LPLCY/LPLA,LCYL
      COMMON/LSHDP/LSTS,LSTD(14)
      COMMON/HITPLT/MPH
      COMMON/GROUND/LGRND,MPXR
      LHIT=.FALSE.
      DHIT=0.
      IF(.NOT.LPLA) RETURN
C!!!  STEP THRU PLATES
      DO 60 MPP=1,MPXR
      MP=MPP
      IF(MP.EQ.MH) GO TO 50
      IF(MH.LT.0) MP=IABS(MH)
C!!!  IF TOTAL SHADOWING ALGORITHM IS BEING USED, HAS PLATE MP
C!!!  SHADOWED EVERY RAY TESTED?
      IF(LSTS.AND..NOT.LSTD(MP)) GO TO 60
      MEX=MEP(MP)
      AN=0.
      DO 5 N=1,3
5     AN=AN+(XIS(N)-X(MP,1,N))*VN(MP,N)
      DN=D(1)*VN(MP,1)+D(2)*VN(MP,2)+D(3)*VN(MP,3)
C!!!  DOES RAY PASS THRU PLATE PLANE?
      IF(AN*DN.GE.0.) GO TO 50
      DO 10 N=1,3
C!!!  CALCULATE POINT WHERE RAY INTERSECTS PLATE PLANE
10    XT(N)=XIS(N)-AN*D(N)/DN
      IF(MP.EQ.MPXR.AND.LGRND)GO TO 11
      DBT=0.
C!!!  IS HIT POINT ON PLATE?
      DO 30 ME=1,MEX
      MME=ME+1
      IF(MME.GT.MEX) MME=1
      RD=0.
      DO 20 N=1,3
20    RD=RD+(X(MP,ME,N)-XT(N))*(X(MP,MME,N)-XT(N))
      CP=VN(MP,1)*((X(MP,ME,2)-XT(2))*(X(MP,MME,3)-XT(3))
     2-(X(MP,ME,3)-XT(3))*(X(MP,MME,2)-XT(2)))
      CP=CP+VN(MP,2)*((X(MP,ME,3)-XT(3))*(X(MP,MME,1)-XT(1))
     2-(X(MP,ME,1)-XT(1))*(X(MP,MME,3)-XT(3)))
      CP=CP+VN(MP,3)*((X(MP,ME,1)-XT(1))*(X(MP,MME,2)-XT(2))
     2-(X(MP,ME,2)-XT(2))*(X(MP,MME,1)-XT(1)))
      DBI=BTAN2(CP,RD)
      DBT=DBT+DBI
30    CONTINUE
      IF(ABS(DBT).LT.PI) GO TO 50
C!!!  CALCULATE DISTANCE TO HIT (DHIT=SHORTEST DHT)
11    DHT=0.
      DO 40 N=1,3
40    DHT=DHT+(XT(N)-XIS(N))*(XT(N)-XIS(N))
      DHT=SQRT(DHT)+1.E-5
      IF(LHIT.AND.(DHT.GT.DHIT)) GO TO 60
```

```
           LHIT=.TRUE.
           DHIT=DH1
           MPH=MP
           IF(MH.GE.Ø) GO TO 6Ø
           DO 45 N=1,3
     C!!!  MOVE HIT POSITION AN INCREMENT TOWARDS SIDE OF PLATE
     C!!!  WHICH SOURCE LIES ON
     45    XIS(N)=XT(N)-SIGN(1.E-5,AN)*VN(MP,N)
           GO TO 61
     5Ø    CONTINUE
           IF(MH.LT.Ø) GO TO 61
     C!!!  IF TOTAL SHADOWING ROUTINE IS BEING USED, INDICATE
     C!!!  THAT PLATE MP DOES NOT SHADOW SOURCE
           IF(LSTS) LSTD(MP)=.FALSE.
     6Ø    CONTINUE
      61   IF (.NOT.LTEST) GO TO 62
           WRITE (6,63)
      63   FORMAT (/,' TESTING PLAINT SUBROUTINE')
           WRITE (6,*) XIS
           WRITE (6,*) D
           WRITE (6,*) DHIT,MH,LHIT
      62   RETURN
           END
```

## POLYRT

PURPOSE

To solve an Mth order polynomial equation.

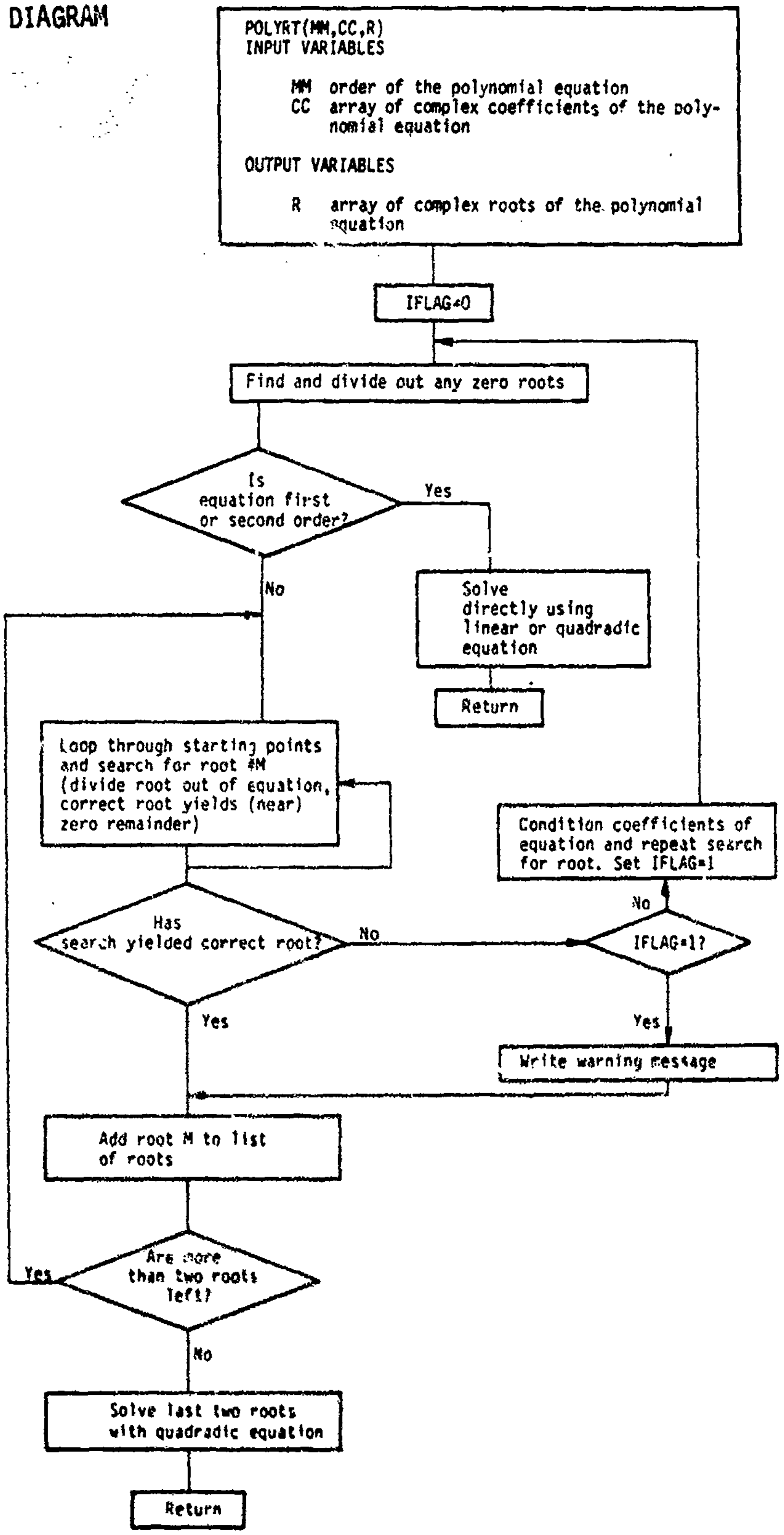
METHOD

This subroutine solves for the roots of an Mth order polynomial,

$$C_M Z^M + C_{M-1} Z^{M-1} + \cdots C_1 Z^1 + C_o = 0.$$

The roots of the polynomial are found using the Newton-Raphson method of iterated synthetic division [16]. The coefficients are stored such that $C_M$ = CC(M+1), $C_o$ = CC(1), etc.

## FLOW DIAGRAM


POLYRT(MM,CC,R)
INPUT VARIABLES
MM order of the polynomial equation
CC array of complex coefficients of the polynomial equation
OUTPUT VARIABLES
R array of complex roots of the polynomial equation
IFLAG=0
Find and divide out any zero roots
Is equation first or second order?
Yes
Solve directly using linear or quadradic equation
Return
No
Loop through starting points and search for root #M (divide root out of equation, correct root yields (near) zero remainder)
Condition coefficients of equation and repeat search for root. Set IFLAG=1
No
Has search yielded correct root?
No
IFLAG=1?
Yes
Yes
Write warning message
Add root M to list of roots
Yes
Are more than two roots left?
No
Solve last two roots with quadradic equation
Return

SYMBOL DICTIONARY

```
C       WORKING ARRAY OF POLYNOMIAL COEFFICIENTS
CC      A COMPLEX ARRAY CONTAINING THE POLYNOMIAL COEFFICIENTS
CMAX    MAGNITUDE OF LARGEST COEFFICIENT
CNEW    ARRAY CONTAINING COEFFICIENTS OF POLYNOMIAL LEFT
        AFTER THE PROSPECTIVE ROOT HAS BEEN FACTORED OUT
CNNW    ARRAY CONTAINING COEFFICIENTS OF POLYNOMIAL LEFT AFTER
        THE PROSPECTIVE ROOT HAS BEEN FACTORED OUT TWICE
EPS     SMALL NUMBER (RELATIVE TO LARGEST COEFFICIENT)
ICONJ   INDEX FOR TRYING THE CONJUGATE OF THE PREVIOUS ROOT
        AS A GUESS
ICOUNT  INDEX ON THE NUMBER OF TIMES THE ITERATION PROCEDURE
        SEARCHES FOR A ROOT
IFLAG   FLAG USED TO INDICATE IF ALL POSSIBLE STARTING VALUES
        HAVE BEEN TRIED
ISTART  INDEX FOR STARTING VALUES
LIMIT   MAXIMUM NUMBER OF ITERATIONS USED TO SEARCH FOR THE ROOT
M       ORDER OF POLYNOMIAL BEING WORKED ON
MI      COMPUTATIONAL VARIABLE


MM      ORDER OF THE EQUATION
MMP1    MM PLUS ONE
MN      ORDER OF ONCE FACTORED POLYNOMIAL BEING WORKED ON
Q       MAGNITUDE OF POLYNOMIAL COEFFICIENTS
R       A COMPLEX ARRAY CONTAINING THE ROOTS OF THE EQUATION
RJ      REMAINDER LEFT AFTER PROSPECTIVE ROOT HAS BEEN
        FACTORED OUT
RJP     REMAINDER LEFT AFTER PROSPECTIVE ROOT HAS BEEN FACTORED
        OUT TWICE
RT      PROSPECTIVE ROOT BEING ITERATED
SR      SQUARE ROOT OF (C(2)*C(2)-4*C(1)*C(3))
START   ARRAY CONTAINING INITIAL GUESS OF ROOT LOCATIONS
TEST    BOUND USED TO DETERMINE IF THE PROSPECTIVE ROOT
        HAS CONVERGED
XI      IMAGINARY PART OF CC
XR      REAL PART OF CC
```

## CODE LISTING

```
C-------------------------------------------------------------------
      SUBROUTINE PCLYRT(MM,CC,R)
C!!!
C!!! THIS ROUTINE SOLVES A COMPLEX POLYNOMIAL EQUATION.
C!!! MM IS THE ORDER OF THE EQUATION
C!!! CC IS A COMPLEX ARRAY CONTAINING THE COEFFICIENTS.
C!!!    CC(1) IS THE CONSTANT TERM, CC(2) THE COEFFICIENT OF Z,
C!!!    ------------,CC(MM+1) THE COEFFICIENT OF Z**MM.
C!!! R  IS A COMPLEX ARRAY IN WHICH THE ROOTS WILL BE RETURNED.
C!!!    IN THE DATA STATEMENT LIMIT IS THE NUMBER OF CYCLES
C!!!    WHICH WILL BE ALLOWED BEFORE THE SEARCH FOR A
C!!!    PARTICULAR ROOT IS TERMINATED. TEST IS THE MAXIMUM
C!!!    INEQUALITY OF THE EQUATION ALLOWED BEFORE A ROOT IS
C!!!    ACCEPTED.
C!!!
      COMPLEX C(21),CC(21),CNEW(21),R(20),SR,RT,Y,DY,RTP
      COMPLEX START(4),CNNN(21),RJ,RJP
      DATA START/(1.,1.),(1.,0.),(-1.,-1.),(-1.,0.)/
      DATA TEST,LIMIT/1.E-05,100/
C!!!  COPY THE INPUT PARAMETERS CC AND MM INTO C AND M.
      IFLAG=0
      MMP1=MM+1
      CMAX=BABS(CC(1))
      DO 9 I=1,MMP1
      C(I)=CC(I)
      IF(BABS(CC(I)).GT.CMAX) CMAX=BABS(CC(I))
9     CONTINUE
      EPS=1.E-5*CMAX
3535  M=MM
      ICONJ=0
C!!!  FIND AND DIVIDE OUT ANY ZERO ROOTS.
2     Q=BABS(C(M+1))
      IF(Q.LT.EPS) GO TO 7
      Q=BABS(C(1))
      IF(Q.GT.EPS) GO TO 1
      DO 8 I=1,M
8     C(I)=C(I+1)
7     R(M)=(0.,0.)
      M=M-1
      IF(M.NE.0) GO TO 2
      RETURN
1     DO 3 N=1,M
3     C(N)=C(N)/C(M+1)
      C(M+1)=(1.,0.)
C!!!  IF EQUATION IS 1ST OR 2ND ORDER SOLVE DIRECTLY AND RETURN.
      IF(M-2) 5,6,4
5     R(1)=-C(1)
      RETURN
C!!!  START SEARCH FOR A ROOT.
4     DO 140 ISTART=1,4
24    RT=START(ISTART)
      IF(ICONJ.EQ.1) RT=CONJG(R(M+1))
      ICOUNT=0
14    CNEW(M)=(1.,0.)
      MN=M-1
      CNNN(MN)=(1.,0.)
      ICOUNT=ICOUNT+1
      IF(ICOUNT.GT.LIMIT) GO TO 141
      DO 111 I=2,M
      MI=M-I
111   CNEW(MI+1)=C(MI+2)+RT*CNEW(MI+2)
      RJ=C(1)+RT*CNEW(1)
      Q=BABS(RJ)
      IF(Q.LE.TEST) GO TO 12
      DO 112 I=2,MN
      MI=MN-I
```

```
  112   CNNW(MI+1)=CNEW(MI+2)+RT*CNNW(MI+2)
        RJP=CNEW(1)+RT*CNNW(1)
        RT=RT-RJ/RJP
        GO TO 14
  141   CONTINUE
        IF(ICONJ.NE.1) GO TO 14Ø
        ICONJ=Ø
        GO TO 24
  14Ø   CONTINUE
        IF(IFLAG.EQ.1) GO TO 15
        IFLAG=1
        DO 9Ø9Ø JJ=1,MMP1
        XR=REAL(CC(JJ))
        XI=AIMAG(CC(JJ))
        IF(ABS(XR).LT.EPS) XR=Ø.
        IF(ABS(XI).LT.EPS) XI=Ø.
        C(JJ)=CMPLX(XR,XI)
  9Ø9Ø  CONTINUE
        GO TO 3535
  15    WRITE(6,16) M,Q
  16    FORMAT(1HØ,4ØH CYCLE LIMIT EXCEEDED WHILE FINDING ROOT,I3,
       218H FINAL INEQUALITY ,F1Ø.4)
  12    CONTINUE
        DO 18 I=1,M
  18    C(I)=CNEW(I)
        R(M)=RT
        M=M-1
        ICONJ=ICONJ+1
        IF(ICONJ.GT.1) ICONJ=Ø
  C!!!  IF MORE THAN TWO ROOTS LEFT RECYCLE THE SEARCH.
        IF(M.GT.2) GO TO 4
  C!!!  FIND THE LAST TWO ROOTS BY THE QUADRATIC FORMULA.
  C     SR=CSQRT(C(2)*C(2)-4.Ø*C(1)*C(3))
        R(1)=(-C(2)+SR)*Ø.5/C(3)
        R(2)=(-C(2)-SR)*Ø.5/C(3)
        RETURN
        END
```

## PRIOUT

### PURPOSE

To output field data in standard format: 4 integer indicators and then magnitude and phase of E-theta and E-phi components.

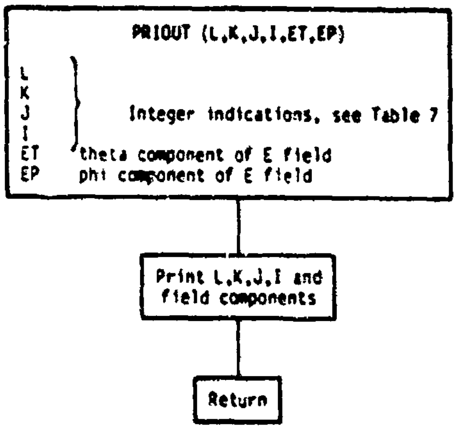
### METHOD

This subroutine is activated by setting LOUT=.TRUE. When the individual field components are being printed out, that is when L≠1000 and when L≠K and J≠I, only the fields with |ET|>0 or |EP|>0 are printed out. A list of the different indicator numbers and what field they correspond to is given in Table 7.

Table 7
Individual field types printed when LOUT=.TRUE.

| L | K | J | I | Field Type |
|---|---|---|---|---|
| 100 | 0 | 0 | 0 | Direct field when plates are present |
| 200 | MP | 0 | 0 | Field reflected from plate MP |
| 300 | MP | MPP | 0 | Field reflected from plate MP then reflected from plate MPP |
| 600 | MP | ME | 0 | Field diffracted from edge ME of plate MP |
| 650 | MP | ME | 0 | Field diffracted from the corners of edge ME of plate MP |
| 700 | MR | MP | ME | Field reflected from plate MR then diffracted from edge ME of plate MP |
| 750 | MR | MP | ME | Field reflected from plate MR then diffracted by the corners of edge ME of plate MP |
| 800 | MP | ME | MR | Field diffracted from edge ME of plate MP then reflected from plate MR |
| 850 | MP | ME | MR | Field diffracted from the corners of edge ME of plate MP then reflected from plate MR |
| 110 | 0 | 0 | 0 | Direct field when only cylinders alone are present |
| 120 | 0 | 0 | 0 | Geometrical optics field reflected by cylinder (for comparison only) |
| 130 | 0 | 0 | 0 | Field scattered by the curved surface of the cylinder |
| 150 | MC | 0 | 0 | Field reflected by end cap MC of the cylinder |
| 500 | MC | 0 | 0 | Field diffracted by the end cap rim MC of the cylinder |
| 240 | MP | 0 | 0 | Geometrical optics field reflected from plate MP then reflected from the curved surface of the,cylinder. (For comparison only) |
| 250 | MP | 0 | 0 | Field reflected from plate MP and then scattered by the curved surface of the cylinder |
| 410 | MP | 0 | 0 | Geometrical optics field reflected from the curved surface of the cylinder and then reflected from plate MP. (For comparison only) |
| 420 | MP | 0 | 0 | Field scattered from the curved surface of the cylinder then reflected from plate MP |
| 940 | MP | ME | 0 | Field reflected from the curved surface of the cylinder then diffracted by edge ME of plate MP |
| 950 | MP | ME | 0 | Field diffracted from edge ME of plate MP then reflected from the curved surface of the cylinder |
| IANGLE | IANGLE | INDEX | INDEX | Sum of fields of a given type (INDEX) for a given angle (IANGLE) |
| 1000 | IANGLE | IANGLE | IANGLE | Total field for a given angle (IANGLE) |

## FLOW DIAGRAM


PRIOUT (L,K,J,I,ET,EP)
L
K
J
I
Integer indications, see Table 7
ET theta component of E field
EP phi component of E field
Print L,K,J,I and field components
Return

## CODE LISTING

```
C------------------------------------------------------------------
      SUBROUTINE PRIOUT(L,K,J,I,ET,EP)
C!!!
C!!!  PRINT OUT DATA IN STANDARD FORMAT.
C!!!  INTEGER INDICATORS,THEN MAG. AND PHASE
C!!!  OF E-THETA&E-PHI COMPONENTS.
C!!!
      COMPLEX ET,EP
      COMMON/FIS/PI,TPI,DPR,RPD
      UTH=CABS(ET)
      UTP=DPR*ATAN2(AIMAG(ET),REAL(ET))
      UPH=CABS(EP)
      UPP=DPR*ATAN2(AIMAG(EP),REAL(EP))
      IF(L.EQ.1000)GO TO 2
      IF(L.EQ.K.AND.J.EQ.I)GO TO 2
      IF(UTH.LT.1.E-5.AND.UPH.LT.1.E-5)RETURN
 2    WRITE(6,1) L,K,J,I,UTH,UTP,UPH,UPP
 1    FORMAT(1H ,4I5,2F15.6,5X,2F15.6)
      RETURN
      END
```

## QFUN

PURPOSE

To compute the q* function for the cylinder's acoustically hard diffraction coefficient.

METHOD

The q* function is defined as[14,15]

$$q^*(x) = \frac{1}{2\sqrt{\pi x}} + \hat{P}_h(x)e^{j\,\pi/4}$$

where

$$\hat{P}_h(x) = \frac{e^{-j\,\pi/4}}{\sqrt{\pi}} \int_{-\infty}^{\infty} \frac{QV(\tau)}{Qw_2(\tau)}\, e^{-jx\tau}\, d\tau$$

and

$V(\tau)$ and $w_2(\tau)$ are Fock type Airy functions,

and

$$Q = \frac{\partial}{\partial\tau}.$$

The q* function is computed as follows:

1) for $x \leq -3$

$$q^*(x) = \frac{1}{2\sqrt{\pi x}} - \frac{1}{2}\sqrt{|x|}\left(1 - j\,\frac{2}{x^3}\right)e^{j\,\frac{x^3}{12}}\, e^{j\,\pi/4} ,$$

2) for $-3 < x < 2$

$$q^*(x) = q^*(x_i) + \frac{(x-x_i)}{(x_{i+1}-x_i)}\,(q^*(x_{i+1})-q^*(x_i)) ,$$

where the $q^*(x_i)$ are tabulated values[14,15] and $x_{i+1}-x_i=0.1$ with $x_i < x < x_{i+1}$.

3) for $x \geq 2$

$$q^*(x) = \frac{1}{2\sqrt{\pi x}} - \frac{e^{j\,\pi/6}}{2\sqrt{\pi}} \sum_{n=1}^{5} \frac{e^{x\bar{q}_n e^{-j\,\frac{5\pi}{6}}}}{\bar{q}_n[A_i(-\bar{q}_n)]}$$

where $A_i(\tau)$ is the Miller type Airy function.

## SYMBOL DICTIONARY

| | |
|---|---|
| AMC | -Ø.5*CEXP(J*PI/6)/SQRT(PI) |
| AQ | MILLER TYPE AIRY FUNCTION AT Q |
| C | Ø.5/SQRT(PI) |
| EXC | CEXP(-5*PI/6) |
| I | SMALLEST INTEGER CLOSEST TO 1Ø*X |
| Q | ZEROES OF DERIVATIVE OF MILLER TYPE AIRY FUNCTION |
| QFUN | Q FUNCTION |
| QI | IMAGINARY PART OF TABULATED Q FUNCTION |
| QR | REAL PART OF TABULATED Q FUNCTION |
| X | ARGUMENT OF Q FUNCTION |
| XI | REAL NUMBER REPRESENTATION OF I |

## CODE LISTING

```
C-----------------------------------------------------------------------
      COMPLEX FUNCTION QFUN(X)
C!!!
C!!!  COMPUTES THE Q FUNCTION OF THE CYLINDER'S
C!!!  DIFFRACTION COEFFICIENT (HARD CASE)
C!!!
      DIMENSION QR(61),QI(61)
      COMPLEX AMC,EXC
      DIMENSION Q(5),AQ(5)
      COMMON/PIS/PI,TPI,DPR,RPD
      DATA AMC,EXC/(-Ø.2443Ø,-Ø.141Ø5),(-Ø.866Ø25,-Ø.5)/
      DATA C/Ø.282Ø9/
      DATA Q/1.Ø1879,3.2482Ø,4.82Ø1Ø,6.16331,7.37218/
      DATA AQ/Ø.53566,-Ø.419Ø2,Ø.38Ø41,-Ø.35791,Ø.3423Ø/
      DATA QR/-.229,-.411,-.559,-.673,-.754,-.8Ø7,-.834,-.841
     2,-.832,-.81Ø,-.78Ø,-.744,-.7Ø5,-.665,-.625,-.587,-.551,
     2-.517,-.486,-.458,-.432,-.4Ø9,-.388,-.369,-.352,-.335,
     2-.32Ø,-.3Ø6,-.293,-.279,-.266,-.253,-.239,-.226,-.212,
     2-.198,-.184,-.17Ø,-.155,-.141,-.126,-.112,-.Ø98,-.Ø84,
     2-.Ø71,-.Ø58,-.Ø46,-.Ø34,-.Ø23,-.Ø12,-.ØØ26,.ØØ64,.Ø15,
     2.Ø22,.Ø29,.Ø36,.Ø41,.Ø46,.Ø51,.Ø56,.Ø61/
      DATA QI/-.838,-.771,-.676,-.562,-.44Ø,-.317,-.199,-.Ø9Ø
     2,.ØØ8,.Ø94,.166,.226,.274,.311,.338,.357,.368,.372,.371
     2,.365,.356,.342,.327,.3Ø9,.289,.268,.246,.223,.2ØØ,.177
     2,.154,.131,.1Ø9,.Ø88,.Ø67,.Ø48,.Ø31,.Ø14,-.ØØ13,-.Ø15
     2,-.Ø27,-.Ø38,-.Ø48,-.Ø56,-.Ø62,-.Ø68,-.Ø72,-.Ø75,-.Ø78
     2,-.Ø79,-.Ø79,-.Ø79,-.Ø78,-.Ø77,-.Ø75,-.Ø72,-.Ø7Ø,-.Ø67
     2,-.Ø64,-.Ø61,-.Ø59/
      IF(X.LE.-3.)GO TO 1
      IF(X.GE.2.)GO TO 2
      I=((3.+X)*1Ø.)
      XI=FLOAT(I)-3Ø.
      I=I+1
      QFUN=CMPLX(QR(I),-QI(I))+(1Ø.*X-XI)*CMPLX(QR(I+1)-QR(I),
     2-QI(I+1)+QI(I))
      RETURN
    1 QFUN=.5*(1./(SQRT(PI)*X)-SQRT(ABS(X))*CEXP(CMPLX(Ø.,Ø.25
     2*PI+X*X*X/12.))*CMPLX(1.,-2./(X*X*X)))
      RETURN
    2 QFUN=(Ø.,Ø.)
      DO 3 N=1,5
    3 QFUN=QFUN+CEXP(Q(N)*X*EXC)/AQ(N)/AQ(N)/Q(N)
      QFUN=QFUN*AMC+C/X
      RETURN
      END
```

## RADCV

### PURPOSE

To compute the longitudinal and transverse radii of curvature of the elliptic cylinder at a given point.

### METHOD

The longitudinal radius of curvature of the elliptic cylinder (in the plane of incidence) at the point defined by elliptical angle VR is given by

$$\rho_g = \frac{(A^2\sin^2 VR+B^2\cos^2 VR)^{3/2}}{AB\ \sin^2\alpha_s} \quad .$$

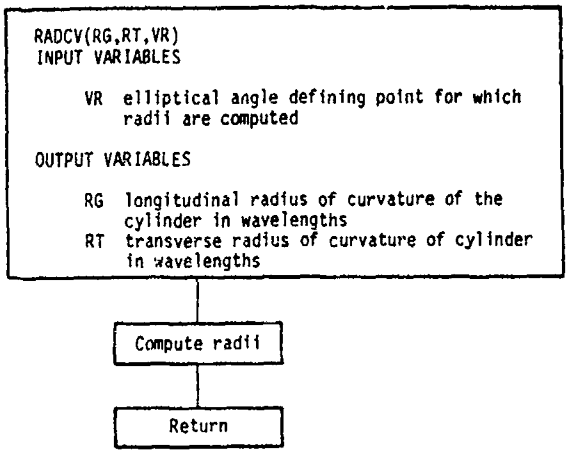
The transverse radius of curvature at the point defined by ell. angle VR is given by

$$\rho_t = \frac{(A^2\sin^2 VR+B^2\cos^2 VR)^{3/2}}{AB\ \sin^2(\alpha_s-\pi/2)} \quad ,$$

where

$\alpha_s$ = AS<br>
$\rho_g$ = RG<br>
$\rho_t$ = RT.

## FLOW DIAGRAM


RADCV(RG,RT,VR)
INPUT VARIABLES
VR elliptical angle defining point for which radii are computed
OUTPUT VARIABLES
RG longitudinal radius of curvature of the cylinder in wavelengths
RT transverse radius of curvature of cylinder in wavelengths
Compute radii
Return

## SYMBOL DICTIONARY

RG    RADIUS OF CURVATURE IN THE PLANE OF INCIDENCE

RGT    RADIUS OF CURVATURE OF THE ELLIPTIC CYLINDER IN THE PRINCIPAL (X-Y) PLANE

RT    RADIUS OF CURVATURE TRANSVERSE TO THE PLANE OF INCIDENCE

VR    ELLIPTIC ANGLE DEFINING THE DESIRED POINT ON CYLINDER

## CODE LISTING

```
C-----------------------------------------------------------------------
      SUBROUTINE RADCV(RG,RT,VR)
C!!!
C!!!  COMPUTES RADII OF CURVATURE OF ELLIPTIC CYLINDER
C!!!
      COMMON/GEOMEL/A,B,ZC(2),SNC(2),CNC(2),CTC(2)
      COMMON/GTD/AS,ID,SAS,SASP,CAS
      DN=SQRT(A*A*SIN(VR)*SIN(VR)+B*B*COS(VR)*COS(VR))
      RGT=DN*DN*DN/A/B
      RG=RGT/SAS/SAS
      IF(SASP.LT.1.E-5)GO TO 1
      RT=RGT/SASP/SASP
      RETURN
    1 RT=1.E20
      RETURN
      END
```

## RCLDPL

### PURPOSE

To compute the far-zone electric field for a source ray which is reflected by the elliptic cylinder and then diffracted by a given edge on a given plate.

### PERTINENT GEOMETRY

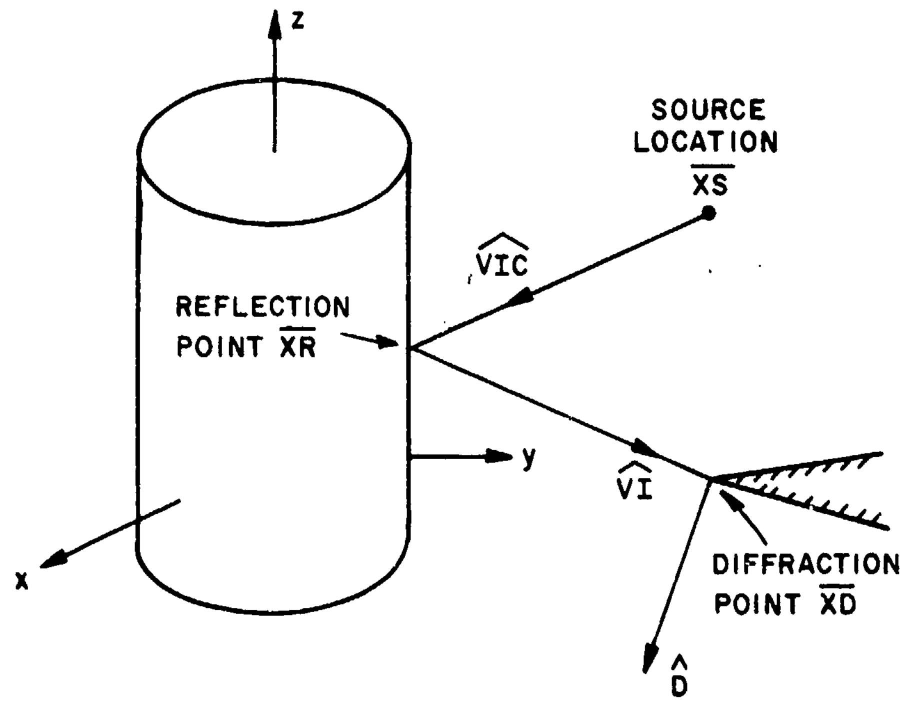

z
SOURCE LOCATION $\overline{XS}$
$\widehat{VIC}$
REFLECTION POINT $\overline{XR}$
y
$\widehat{VI}$
x
DIFFRACTION POINT $\overline{XD}$
$\hat{D}$

Figure 86--Ray reflected by cylinder and then diffracted by plate edge.

### METHOD

The field reflected by the elliptic cylinder and then diffracted by a plate edge is calculated in this subroutine. The field reflected by the cylinder is found using geometrical optics[4]. This causes an astigmatic tube of rays to be inciaent on the plate edge. The unifurm Geometrical Theory of Diffraction[4] is then used to find the diffracted field from the edge. The resultant field in the far zone has the form (pp. 154-155, Reference 1)

$$\vec{E}^{r,d} = \vec{E}^{i}(Q_R) \cdot \overline{\overline{R}} \cdot \overline{\overline{D}} \sqrt{\frac{\rho_1^r \rho_2^r}{(\rho_1^r+s')(\rho_2^r+s')}} \sqrt{\rho_e^i}\, e^{-jks'} \frac{e^{-jks}}{s} ,$$

where $\vec{E}^i(Q_R)$ is the incident field at the reflection point $Q_R$, $\overline{\overline{R}}$ is the diadic reflection coefficient, $\overline{\overline{D}}$ is the dyadic edge diffraction coefficient, $\rho_1^r$ and $\rho_2^r$ are the reflected ray caustic distances, $\rho_e^i$ is the incident caustic distance on the edge, s' is the distance from the reflection point to the diffraction point, and s is the distance from the diffraction point in the far zone. The geometry is shown in Figure 85 and further illustrations can be found in the write ups for subroutines REFCYL and DIFPLT. The phase of the field is referred to the reference coordinate system origin so that

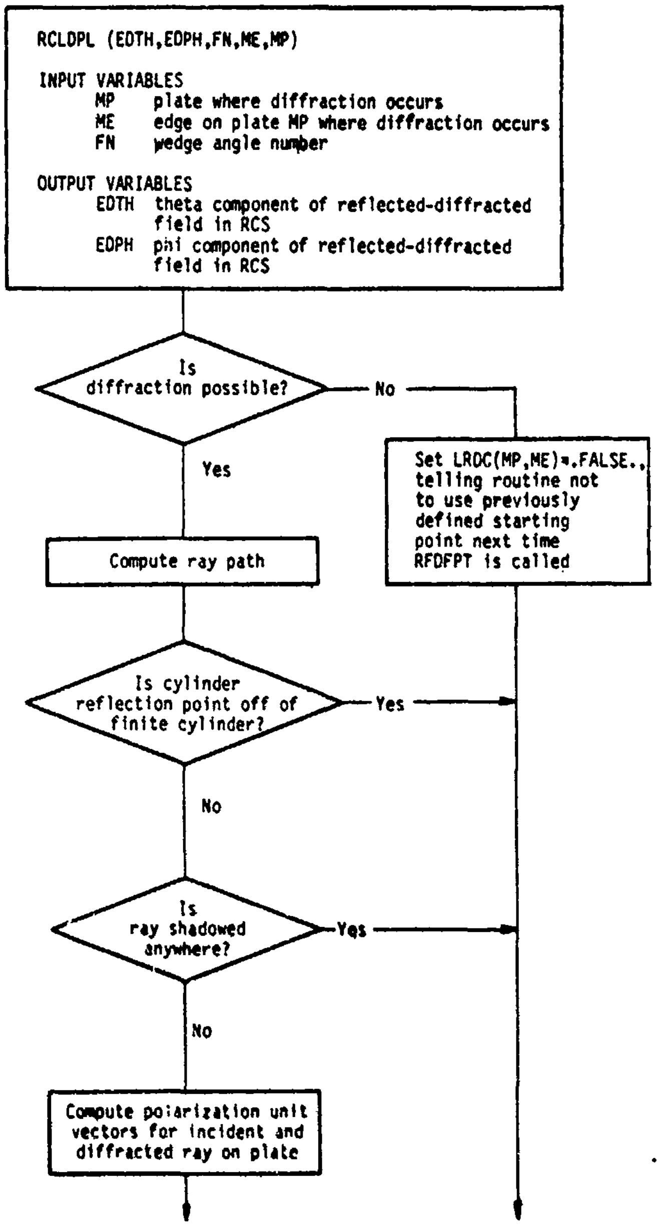
$$\frac{e^{-jks}}{s} = e^{jk\hat{D}\cdot\overline{X}_d} \frac{e^{-jkR}}{R} .$$

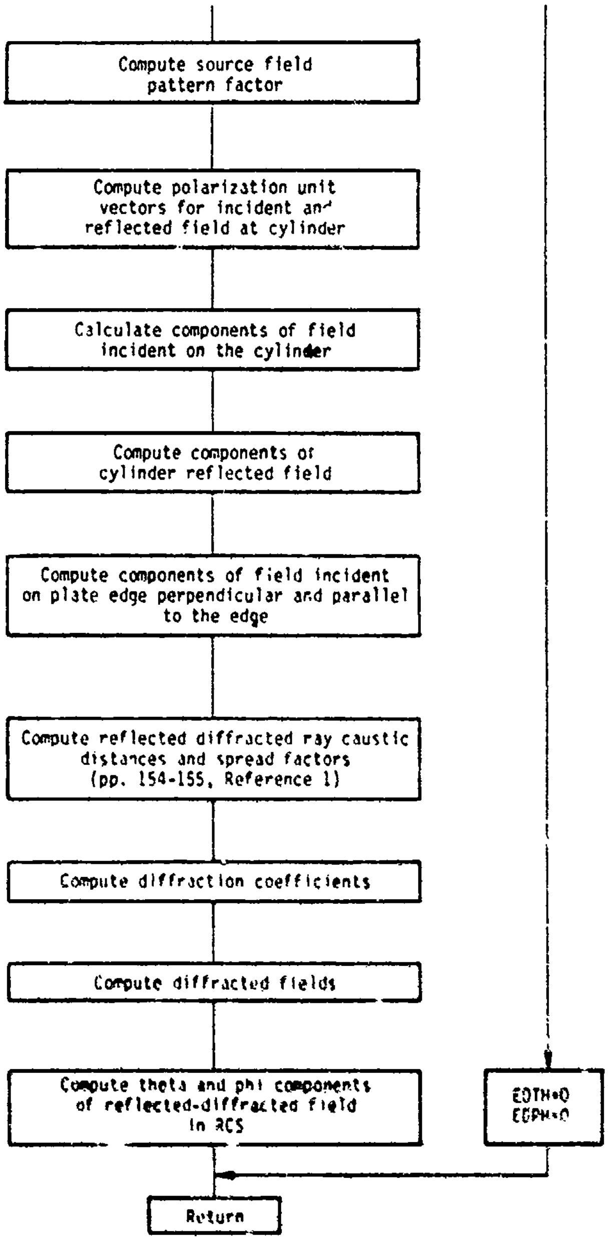
The reflected-diffracted field then has the form

$$\vec{E}^{r,d} = W_m(EDTH\hat{\theta}+EDPH\hat{\phi}) \frac{e^{-jkR}}{R} ,$$

where the factor $\frac{e^{-jkR}}{R}$ and the source weight $(W_m)$ are added elsewhere in the code.

## FLOW DIAGRAM


RCLDPL (EDTH,EDPH,FN,ME,MP)
INPUT VARIABLES
MP plate where diffraction occurs
ME edge on plate MP where diffraction occurs
FN wedge angle number
OUTPUT VARIABLES
EDTH theta component of reflected-diffracted field in RCS
EDPH phi component of reflected-diffracted field in RCS
Is diffraction possible?
No
Yes
Set LROC(MP,ME)=.FALSE., telling routine not to use previously defined starting point next time RFDFPT is called
Compute ray path
Is cylinder reflection point off of finite cylinder?
Yes
No
Is ray shadowed anywhere?
Yes
No
Compute polarization unit vectors for incident and diffracted ray on plate


Compute source field pattern factor
Compute polarization unit vectors for incident and reflected field at cylinder
Calculate components of field incident on the cylinder
Compute components of cylinder reflected field
Compute components of field incident on plate edge perpendicular and parallel to the edge
Compute reflected diffracted ray caustic distances and spread factors (pp. 154-155, Reference 1)
Compute diffraction coefficients
Compute diffracted fields
Compute theta and phi components of reflected-diffracted field in RCS
EDTH=0
EDPH=0
Return

## SYMBOL DICTIONARY

| | |
|---|---|
| BO | DIFFRACTED FIELD POLARIZATION UNIT VECTOR PARALLEL TO EDGE |
| BOP | INCIDENT FIELD POLARIZATION UNIT VECTOR PARALLEL TO EDGE |
| DD | NORMALIZATION CONSTANT FOR CYLINDER TANGENT VECTOR |
| DH | EDGE DIFFRACTION COEFFICIENT FOR HARD FIELD COMPS. |
| DHIT | DISTANCE FROM SOURCE TO HIT POINT (FROM PLAINT) |
| DOTP | TEST PARAMETER USED TO DETERMINE IF REFL IS LEGAL |
| DS | DIFFRACTION COEFFICIENT FOR SOFT FIELD COMPONENTS |
| DV | DOT PRODUCT OF EDGE UNIT VECTOR AND DIFFRACTED RAY PROPAGATION DIRECTION |
| EDPH | PHI COMPONENT OF DIFFRACTED FIELD IN RCS |
| EDPL | DIFFRACTED FIELD COMPONENT PARALLEL TO EDGE |
| EDPR | DIFFRACTED FIELD COMPONENT PERPENDICULAR TO EDGE |
| EDTH | THETA COMPONENT OF DIFFRACTED FIELD IN RCS |
| EIPL | COMPONENT OF FIELD INCIDENT ON CYLINDER (OR PLATE) PARALLEL TO PLANE OF INCIDENCE (OR EDGE) |
| EIPR | COMPONENT OF FIELD INCIDENT ON CYLINDER (OR PLATE) PERPENDICULAR TO PLANE OF INCIDENCE (OR EDGE) |
| EIX<br>EIY<br>EIZ | SOURCE PATTERN FACTORS FOR X,Y,Z COMPONENTS OF INCIDENT E-FIELD |
| ERX<br>ERY<br>ERZ | X,Y,Z COMPONENTS OF CYLINDER REFLECTED FIELD IN RCS |
| EXPH | COMPLEX PHASE AND SPREADING FACTOR |
| LDRC | SET TRUE IF REFL DATA IS AVAILABLE FROM PREVIOUS PATTERN ANGLE (OR FOR NEXT PATTERN ANGLE (WHEN LEAVING ROUTINE)) |
| LHIT | SET TRUE IF RAY HITS PLATE (FROM PLAINT) |
| ME | EDGE ON PLATE MP WHERE DIFFRACTION OCCURS |
| MP | PLATE WHERE DIFFRACTION OCCURS |
| PH | DIFFRACTED FIELD POLARIZATION UNIT VECTOR NORMAL TO EDGE |
| PHICR | PHI COMPONENT OF FIELD INCIDENT ON CYLINDER IN RCS |
| PHO | INCIDENT FIELD POLARIZATION UNIT VECTOR NORMAL TO EDGE |
| PSOR | INCIDENT RAY PHI ANGLE IN DIFFRACTION POINT COORD SYS |
| PSR | DIFFRACTED RAY PHI ANGLE IN DIFFRACTION POINT COORD SYS |
| RHI1 | CAUSTIC DISTANCE OF CYLINDER REFLECTED FIELD INCIDENT ON EDGE IN THE DIRECTION PERPENDICULAR TO THE EDGE |
| RHI2 | CAUSTIC DISTANCE OF CYLINDER REFLECTED FIELD INCIDENT ON EDGE IN THE DIRECTION PARALLEL TO THE EDGE |
| RHIE | EDGE CAUSTIC DISTANCE |
| RHO1 | RAY SPREADING RADIUS AT CYLINDER IN PLANE NORMAL TO PLANE OF INCIDENCE |
| RHO2 | RAY SPREADING RADIUS AT CYL IN PLANE OF INCIDENCE |
| SMAG | LENGTH OF RAY FROM REFL POINT ON CYL TO SOURCE |
| SP | DISTANCE BETWEEN REFLECTION AND DIFFRACTION POINT |
| THICR | THETA COMPONENT OF INCIDENT RAY DIRECTION ON CYLINDER IN RCS |
| TPP | DISTANCE PARAMETER FOR EDGE DIFFRACTED FIELD |
| UIPPX<br>UIPPY<br>UIPPZ | X,Y,Z COMPONENTS OF INCIDENT POLARIZATION UNIT VECTOR PARALLEL TO PLANE OF INCIDENCE |
| UIPRX<br>UIPRY<br>UIPRZ | X,Y,Z COMPONENTS OF INC/REFL POLARIZATION UNIT VECTOR PERPENDICULAR TO PLANE OF INCIDENCE |
| URPPX<br>URPPY<br>URPPZ | X,Y,Z COMPONENTS OF REFLECTED POLARIZATION UNIT VECTOR PARALLEL TO THE PLANE OF INCIDENCE |
| VI | X,Y,Z COMPONENTS OF RAY PROPAGATION DIRECTION OF RAY INCIDENT ON DIFFRACTION POINT |
| VIC | X,Y,Z COMPONENTS OF RAY PROPAGATION DIRECTION OF RAY INCIDENT ON CYLINDER |

| | |
|---|---|
| VR | ELL ANGLE DEFINING REFLECTION POINT ON CYL (2-D) |
| XD | X,Y,Z COMPONENTS OF DIFFRACTION POINT IN RCS |
| XDP | MODIFIED DIFFRACTION POINT LOCATION FOR SHADOWING TEST |
| XR | X,Y,Z COMPONENTS OF REFLECTION POINT ON CYL |

# CODE LISTING

```
C-----------------------------------------------------------------
      SUBROUTINE RCLDPL(EDTH,EDPH,FN,ME,MP)
C!!!
C!!!  COMPUTES THE FIELD REFLECTED FROM THE ELLIPTIC CYLINDER
C!!!  THEN DIFFRACTED FROM EDGE #ME OF PLATE #MP
C!!!
      COMPLEX EF,EG,EIPR,EIPL,EXPH,DS,DH,DPS,DPH,EDPR,EDPL,EDTH,EDPH
      COMPLEX ERFR,ERPP,EIX,EIY,EIZ,ERX,ERY,ERZ
      DIMENSION UN(2),UB(2),VIC(3),XR(3)
      DIMENSION VI(3),XD(3),PHO(3),PH(3),BOP(3),BO(3),XDP(3)
      LOGICAL LHIT,LRDC,LDEBUG,LTEST
      COMMON/GEOPLA/X(14,6,3),V(14,6,3),VP(14,6,3),VN(14,3)
     2,MEP(14),MPX
      COMMON/SORINF/XS(3),VXS(3,3)
      COMMON/DIR/D(3),THSR,PHSR,SPHS,CPHS,STHS,CTHS
      COMMON/GEOMEL/A,B,ZC(2),SNC(2),CNC(2),CTC(2)
      COMMON/ENDRCL/VCD(14,6),UCD(14,6),BCD(14,6,2)
      COMMON/THPHUV/DT(3),DP(2)
      COMMON/PIS/PI,TPI,DPR,RPD
      COMMON/TEST/LDEBUG,LTEST
      COMMON/CLRDC/LRDC(14,6)
      IF(FN.GT.2.) GO TO 40
      DV=0.
      DO 10 N=1,3
   10 DV=DV+D(N)*V(MP,ME,N)
C!!!  IS DIFFRACTION POSSIBLE?
      IF(DV.LT.BCD(MP,ME,1).OR.DV.GT.BCD(MP,ME,2)) GO TO 39
C!!!  COMPUTE RAY PATH
      CALL RFDFPT(VR,XR,DOTP,DD,SMAG,VIC,XD,SP,VI,DV,ME,MP
     2,LRDC(MP,ME))
C!!!  IS REFLECTION LEGAL?
      IF(DOTP.LE.0.) GO TO 40
C!!!  IS REFLECTION POINT OFF OF FINITE CYLINDER?
      IF(XR(3).GT.ZC(1)+XR(1)*CTC(1).OR.
     2XR(3).LT.ZC(2)+XR(1)*CTC(2)) GO TO 40
      CNP=COS(FN*0.5*PI)
      SNP=SIN(FN*0.5*PI)
      DO 10 N=1,3
      VECT=VP(MP,ME,N)*CNP+VN(MP,N)*SNP
   10 XDP(N)=XD(N)+VECT*1.E-5
C!!!  IS DIFFRACTED RAY SHADOWED BY A PLATE OR
C!!!  A CYLINDER?
      CALL PLAINT(XDP,D,DHIT,MP,LHIT)
      IF(LHIT) GO TO 40
      CALL CYLINT(XD,D,PPSR,DHIT,LHIT,.TRUE.)
      IF(LHIT) GO TO 40
C!!!  IS RAY BETWEEN REFLECTION AND DIFFRACTION SHADOWED?
      CALL PLAINT(XR,VI,DHIT,MP,LHIT)
      IF(LHIT.AND.(DHIT.LT.SP)) GO TO 40
C!!!  IS RAY INCIDENT ON CYLINDER SHADOWED?
      CALL PLAINT(XS,VIC,DHIT,0,LHIT)
      IF(LHIT.AND.(DHIT.LT.SMAG)) GO TO 40
      QI=0.
      PP=0.
      QQ=0.
      PQ=0.
      DO 20 N=1,3
      QI=QI-VN(MP,N)*VI(N)
      PP=PP-VP(MP,ME,N)*VI(N)
      QQ=QQ+VN(MP,N)*D(N)
   20 PQ=PQ+VP(MP,ME,N)*D(N)
      PSOM=ATAN2(QI,PP)
      PSO=TPI+PSOM
      IF(PSO.LT.0.) PSO=360.+PSO
      PSN=ATAN2(QQ,PQ)
      PSN=TPI+PSN
      IF(PS.LT.0.) PS=360.+PS
      FNP=FN*180.+1.E-4
      IF(PSO.GT.FNP.OR.PS.GT.FNP) GO TO 40
      SPHO=SIN(PSOR)
      CPHO=COS(PSOR)
      SPH=SIN(PSR)
      CPH=COS(PSR)
C!!!  COMPUTE POLARIZATION UNIT VECTORS FOR INCIDENT
C!!!  AND DIFFRACTED FIELD ON PLATE
      DO 30 N=1,3
      PHO(N)=-VP(MP,ME,N)*SPHO+VN(MP,N)*CPHO
30    PH(N)=-VP(MP,ME,N)*SPH+VN(MP,N)*CPH
      BOP(1)=PHO(2)*VI(3)-PHO(3)*VI(2)
      BOP(2)=PHO(3)*VI(1)-PHO(1)*VI(3)
      BOP(3)=PHO(1)*VI(2)-PHO(2)*VI(1)
      BO(1)=PH(2)*D(3)-PH(3)*D(2)
      BO(2)=PH(3)*D(1)-PH(1)*D(3)
      BO(3)=PH(1)*D(2)-PH(2)*D(1)
      THICR=BTAN2(SQRT(VIC(1)*VIC(1)+VIC(2)*VIC(2)),VIC(3))
      PHICR=BTAN2(VIC(2),VIC(1))
      CALL SOURCE(EF,EG,EIX,EIY,EIZ,THICR,PHICR,VXS)
      RG=DD*DD*DD/A/B
      CALL NANDB(UN,UB,VR)
      CTHC=UN(1)*VI(1)+UN(2)*VI(2)
      RR=BTAN2(-VIC(1)*UB(1)-VIC(2)*UB(2),-VIC(3))
      SR=SIN(RR)
      CR=COS(RR)
      SST2=SR*SR+CR*CR*CTHC*CTHC
      RHO2=SMAG
      RHO1=SMAG*RG*CTHC/(RG*CTHC+2.*SMAG*SST2)
C!!!  COMPUTE POLARIZATION UNIT VECTORS FOR INCIDENT
C!!!  AND REFLECTED FIELDS AT CYLINDER
      UIPRX=SIN(RR-.5*PI)*UB(1)
      UIPRY=SIN(RR-.5*PI)*UB(2)
      UIPRZ=COS(RR-.5*PI)
      UIPPX=VIC(3)*UIPRY-VIC(2)*UIPRZ
      UIPPY=VIC(1)*UIPRZ-VIC(3)*UIPRX
      UIPPZ=VIC(2)*UIPRX-VIC(1)*UIPRY
      URPPX=VI(3)*UIPRY-VI(2)*UIPRZ
      URPPY=VI(1)*UIPRZ-VI(3)*UIPRX
      URPPZ=VI(2)*UIPRX-VI(1)*UIPRY
C!!!  CALCULATE COMPONENTS OF FIELD INCIDENT ON CYLINDER
C!!!  PERPENDICULAR AND PARALLEL TO PLANE OF INCIDENCE
      EXPH=CEXP(CMPLX(0.,-TPI*SMAG))/SMAG
      EIPR=UIPRX*EIX+UIPRY*EIY+UIPRZ*EIZ
      EIPL=UIPPX*EIX+UIPPY*EIY+UIPPZ*EIZ
C!!!  COMPUTE COMPONENTS OF CYLINDER REFLECTED FIELD
      ERPR=-SQRT(RHO1*RHO2)*EXPH*EIPR
      ERPP=SQRT(RHO1*RHO2)*EXPH*EIPL
      ERX=ERPR*UIPRX+ERPP*URPPX
      ERY=ERPR*UIPRY+ERPP*URPPY
      ERZ=ERPR*UIPRZ+ERPP*URPPZ
C!!!  COMPUTE COMPONENTS OF FIELD INCIDENT ON PLATE
C!!!  EDGE PARALLEL AND PERPENDICULAR TO EDGE
      EIPR=ERX*PHO(1)+ERY*PHO(2)+ERZ*PHO(3)
      EIPL=ERX*BOP(1)+ERY*BOP(2)+ERZ*BOP(3)
      SDO=SQRT((VP(MP,ME,3)*D(2)-VP(MP,ME,2)*D(3))**2+(VP(MP,ME,1)
     2*D(3)-VP(MP,ME,3)*D(1))**2+(VP(MP,ME,2)*D(1)-VP(MP,ME,1)*D(2))
     2**2)
C!!!  COMPUTE REFLECTED-DIFFRACTED RAY CAUSTIC
C!!!  DISTANCES AND SPREAD FACTORS
      UIN1=UN(1)*UIPPX+UN(2)*UIPPY
      UIN2=UN(1)*UIPRX+UN(2)*UIPRY
      URN1X=UIPPX-2.*UIN1*UN(1)
      URN1Y=UIPPY-2.*UIN1*UN(2)
      URN1Z=UIPPZ
      UXR2X=UIPRX-2.*UIN2*UN(1)
      UXR2Y=UIPRY-2.*UIN2*UN(2)
      UXR2Z=UIPRZ
      TH11=UIPPX*UB(1)+UIPPY*UB(2)
      TH12=UIPPZ
      TH21=UIPRX*UB(1)+UIPRY*UB(2)
      TH22=UIPRZ
      DET=TH11*TH22-TH12*TH21
      QR11=1./SMAG+2.*CTHC*TH22*TH22/(RO*DET*DET)
      QR12=-2.*CTHC*TH22*TH12/(RO*DET*DET)
      QR22=1./SMAG+2.*CTHC*TH12*TH12/(RO*DET*DET)
      QRH=QR22-1./RHO1
      QDH=SQRT(QRH*QRH+QR12*QR12)
      XR1X=(QRH*UXR1X-QR12*UXR2X)/QDH
      XR1Y=(QRH*UXR1Y-QR12*UXR2Y)/QDH
      XR1Z=(QRH*UXR1Z-QR12*UXR2Z)/QDH
      XR2X=-(VI(2)*XR1Z-VI(3)*XR1Y)
      XR2Y=-(VI(3)*XR1X-VI(1)*XR1Z)
      XR2Z=-(VI(1)*XR1Y-VI(2)*XR1X)
      CXR1=V(MP,ME,1)*XR1X+V(MP,ME,2)*XR1Y+V(MP,ME,3)*XR1Z
      CXR2=V(MP,ME,1)*XR2X+V(MP,ME,2)*XR2Y+V(MP,ME,3)*XR2Z
      RHIE=RHO1*RHO2/(RHO2*CXR1*CXR1+RHO1*CXR2*CXR2)
      RHIE=RHIE+SP
      RHI1=RHO1+SP
      RHI2=RHO2+SP
      TPP=RHI1*RHI2*SBO*SBO/RHIE
      GAM=XD(1)*D(1)+XD(2)*D(2)+XD(3)*D(3)
      EXPH=CEXP(CMPLX(0.,TPI*(GAM-SP)))/SQRT(RHI1*RHI2)
      EXPH=EXPH*SQRT(RHIE)
C!!!  COMPUTE DIFFRACTION COEFFICIENTS
      CALL UR(DS,DH,DPS,DPH,TPP,PS,PSO,SBO,FN,.FALSE.)
C!!!  COMPUTE DIFFRACTED FIELDS
      EDPH=-EIPH*DH*EXPH
      EDPL=-EIPL*DS*EXPH
C!!!  COMPUTE THETA AND PHI COMPONENTS OF DIFFRACTED
C!!!  FIELD IN PCS
      EDTH=EDPL*(BO(1)*DT(1)+BO(2)*DT(2)+BO(3)*DT(3))
     2+EDPH*(PH(1)*DT(1)+PH(2)*DT(2)+PH(3)*DT(3))
      EDPH=EDPL*(BO(1)*DP(1)+BO(2)*DP(2))
     2+EDPH*(PH(1)*DP(1)+PH(2)*DP(2))
      GO TO 999
 39   LRDC(MP,ME)=.FALSE.
 40   CONTINUE
      EDTH=(0.,0.)
      EDPH=(0.,0.)
 999  CONTINUE
      IF(.NOT.LTEST) RETURN
      WRITE(6,991)
 991  FORMAT(/,' TESTING RCLDPL SUBROUTINE')
      WRITE(6,*) EDTH,EDPH,FN,ME,MP
      RETURN
      END
```

## RCLRPL

### PURPOSE

To calculate the geometrical optics fields of a source ray which is reflected by the elliptic cylinder and then reflected by a given plate.

### PERTINENT GEOMETRY

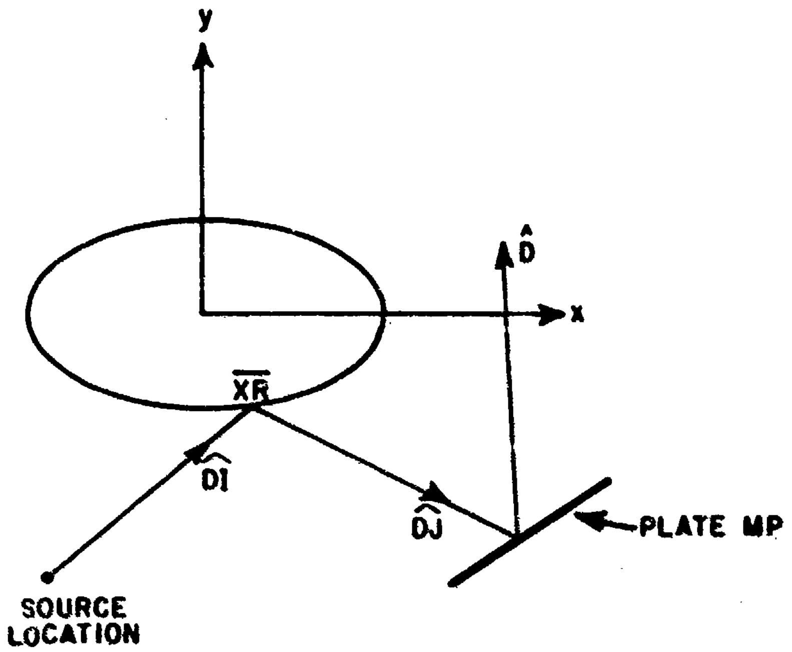


Figure 87--Illustration of ray reflected by cylinder and then reflected by a plate.

### METHOD

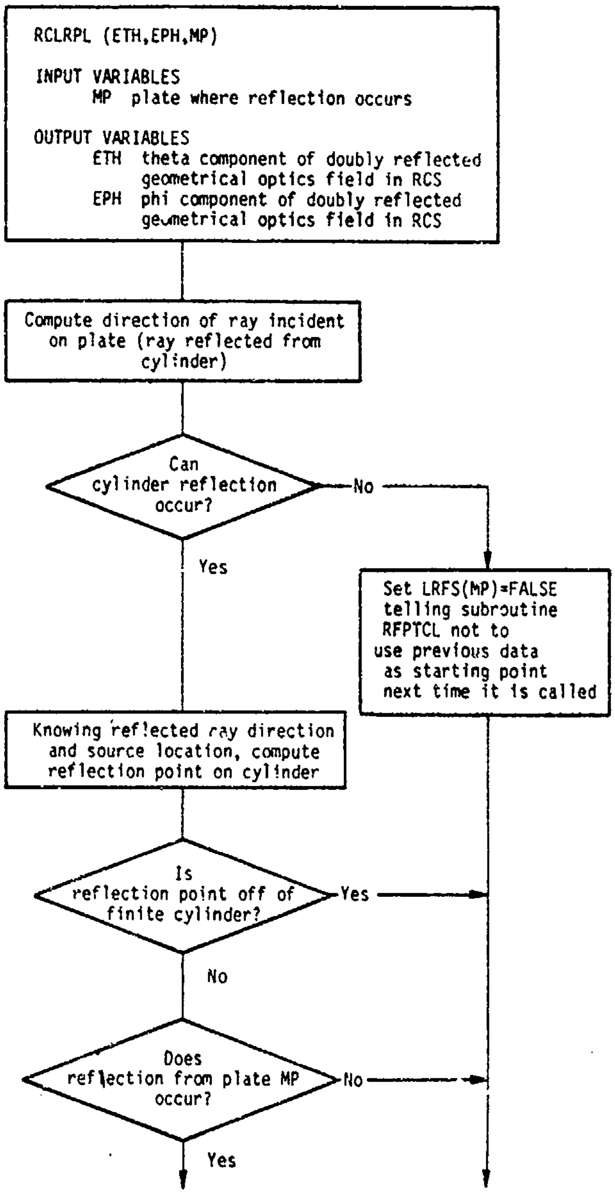
Subroutine RCLRPL functions as a service routine for subroutine SCLRPL, where the actual cylinder fields are computed. The geometrical optics reflected field components ETH and EPH computed in RCLRPL are used only for reference purposes (when LOUT is set true). The field components calculated in RCLRPL which are used in SCLRPL are the hard and soft components of the source field incident on the cylinder at the reflection point. These components, along with several other useful parameters are passed to subroutine SCLRPL through common block FUDGJ.

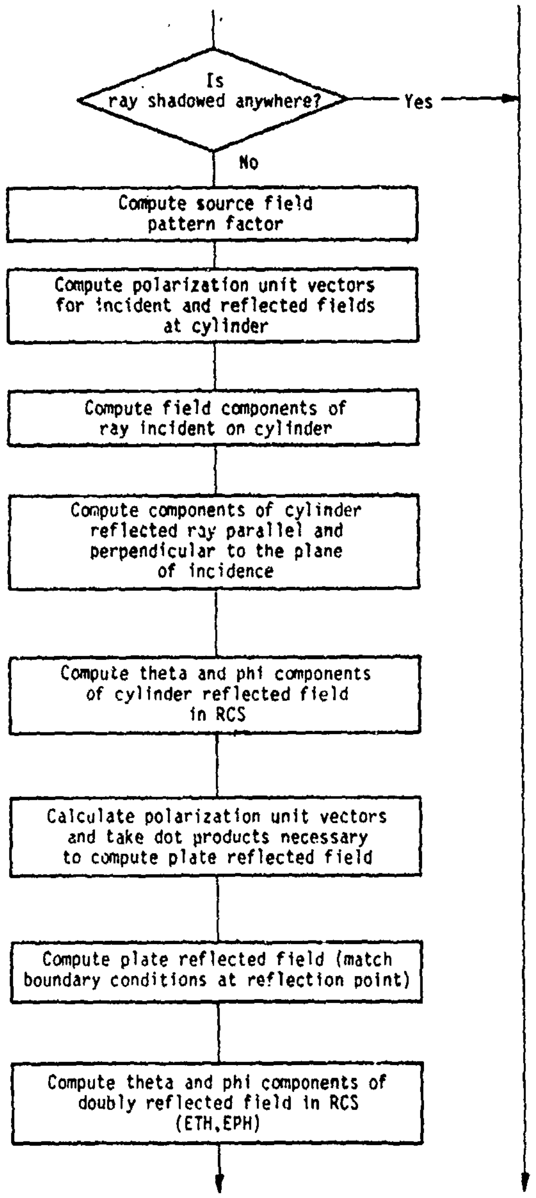
The geometrical optics fields determined in this subroutine, for the reflection from the cylinder, are calculated in the direction DJ. This direction is found by imaging the observation direction into the plate, as illustrated in Figure 87. The cylinder reflected fields are found in a similar manner to those obtained in subroutine REFCYL. The plate reflected fields are found by satisfying the boundary conditions for the fields on the surface of the plate. The phase of the resultant double reflected field is referred to the reference coordinate system origin. The double reflected field thus has the form

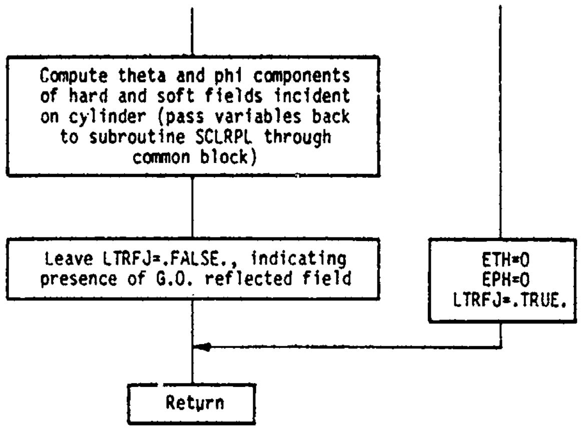
$$\vec{E}^{r,r} = W_m(\mathrm{ETH}\hat{\theta}+\mathrm{EPH}\hat{\phi}) \frac{e^{-jkR}}{R} ,$$

where the factor $\frac{e^{-jkR}}{R}$ and the source weight ($W_m$) are added elsewhere in the code.

## FLOW DIAGRAM


RCLRPL (ETH,EPH,MP)
INPUT VARIABLES
MP plate where reflection occurs
OUTPUT VARIABLES
ETH theta component of doubly reflected geometrical optics field in RCS
EPH phi component of doubly reflected geometrical optics field in RCS
Compute direction of ray incident on plate (ray reflected from cylinder)
Can cylinder reflection occur?
No
Yes
Set LRFS(MP)=FALSE telling subroutine RFPTCL not to use previous data as starting point next time it is called
Knowing reflected ray direction and source location, compute reflection point on cylinder
Is reflection point off of finite cylinder?
Yes
No
Does reflection from plate MP occur?
No
Yes


Is
ray shadowed anywhere?
Yes
No
Compute source field pattern factor
Compute polarization unit vectors for incident and reflected fields at cylinder
Compute field components of ray incident on cylinder
Compute components of cylinder reflected ray parallel and perpendicular to the plane of incidence
Compute theta and phi components of cylinder reflected field in RCS
Calculate polarization unit vectors and take dot products necessary to compute plate reflected field
Compute plate reflected field (match boundary conditions at reflection point)
Compute theta and phi components of doubly reflected field in RCS (ETH,EPH)


Compute theta and phi components of hard and soft fields incident on cylinder (pass variables back to subroutine SCLRPL through common block)
Leave LTRFJ=.FALSE., indicating presence of G.O. reflected field
ETH=0
EPH=0
LTRFJ=.TRUE.
Return

# SYMBOL DICTIONARY

| | |
|---|---|
| A1<br>A2<br>A3 | FIELD COMPONENTS OF RAY INCIDENT ON PLATE NORMAL AND TANGENT TO THE PLATE DETERMINANT OF POLARIZATION TRANSFORMATION |
| C11<br>C12<br>C21<br>C22 | COEFFICIENTS USED TO CONVERT POLARIZATION FROM THETA AND PHI COMPONENTS IN RCS TO COMPONENTS NORMAL AND TANGENT TO PLATE (AND VICE-VERSA) |
| D | PROPAGATION DIRECTION AFTER PLATE REFL. IN (X,Y,Z) RCS COMPONENTS |
| DD1 | DOT PRODUCT OF UNIT VECTOR OF PROPAGATION DIRECTION AND CYLINDER TANGENT UNIT VECTOR THROUGH TAN POINT 1 (2-D) |
| DD2 | DOT PRODUCT OF UNIT VECTOR OF PROPAGATION DIRECTION AND CYLINDER TANGENT UNIT VECTOR THROUGH TAN POINT 2 (2-D) |
| DHIT | DISTANCE FROM SOURCE TO HIT POINT (FROM PLAINT) |
| DH1 | DISTANCE TO HIT POINT (FROM PLAINT AND CYLINT) |
| DI | X,Y, AND Z COMPONENTS OF INCIDENT RAY DIRECTION ON CYL IN RCS |
| DJ | X,Y,Z COMPONENTS OF PROPAGATION DIRECTION OF RAY INCIDENT ON PLATE |
| EF | PATTERN FACTOR OF THETA COMPONENT OF INCIDENT FIELD IN RCS (ALSO THETA COMPONENT OF CYL REFLECTED FIELD IN RCS) |
| EG | PATTERN FACTOR OF PHI COMPONENT OF INCIDENT FIELD IN RCS (ALSO PHI COMPONENT OF CYL REFL FIELD IN RCS) |
| EHPH | PHI COMPONENT OF HARD COMPONENT OF FIELD INCIDENT ON CYLINDER |
| EHTH | THETA COMPONENT OF THE HARD COMPONENT OF FIELD INC ON CYL (PARALLEL TO PLANE OF INCIDENCE) |
| EIPP | INCIDENT FIELD COMPONENT PARALLEL TO PLANE OF INCIDENCE ON CYLINDER |
| EIPR | INCIDENT FIELD COMPONENT PERPENDICULAR TO PLANE OF INC ON CYL |
| ERPP | COMPONENT OF CYLINDER REFLECTED FIELD PARALLEL TO PLANE OF INCIDENCE |
| ERPR | COMPONENT OF CYLINDER REFLECTED FIELD PERPENDICULAR TO PLANE OF INCIDENCE |
| ERX<br>ERY<br>ERZ | X,Y,Z COMPONENTS OF CYLINDER REFLECTED FIELD IN RCS |
| ESPH | PHI COMPONENT OF SOFT COMPONENT OF FIELD INCIDENT ON CYL |
| ESTH | THETA COMPONENT OF SOFT COMPONENT OF FIELD INCIDENT ON CYL |
| EX<br>EY<br>EZ | X,Y,Z COMPONENTS OF SOURCE FIELD PATTERN FACTOR IN RCS |
| GAM | PHASE CONSTANT |
| LHIT | SET TRUE IF RAY HITS PLATE (FROM PLAINT) |
| LRFS | SET TRUE IF REFL DATA IS AVAILABLE FROM PREVIOUS PATTERN ANGLE (OR FOR NEXT PATTERN ANGLE (WHEN LEAVING ROUTINE)) |
| LTRFJ | SET TRUE IF G.O. REFLECTED-REFLECTED FIELDS DO NOT EXIST |
| PH | COMPLEX PHASE CONSTANT |
| PHIR | PHI COMPONENT OF INCIDENT RAY DIRECTION ON CYL IN RCS |
| PHJR | PHI COMPONENT OF RAY PROPAGATION DIRECTION BETWEEN CYLINDER AND PLATE IN RCS |
| RHO1 | RAY SPREADING RADIUS IN PLANE OF CYLINDER CURVATURE AT REFLECTION POINT |
| RHO2 | RAY SPREADING RADIUS IN PLANE NORMAL TO PLANE OF INCIDENCE AT REFLECTION POINT |
| SMAG | LENGTH OF RAY FROM REFL POINT ON CYL TO SOURCE |
| SXN<br>SYN<br>SZN | X,Y,Z COMPONENTS OF UNIT VECTOR OF RAY FROM REFL. POINT ON CYLINDER TO SOURCE LOCATION IN RCS |
| THIR | THETA COMPONENT OF INCIDENT RAY DIRECTION ON CYLINDER |
| THJR | THETA COMPONENT OF RAY PROPAGATION DIRECTION BETWEEN CYLINDER AND PLATE |

| | |
|---|---|
| UIPPX<br>UIPPY<br>UIPPZ | X,Y,Z COMPONENTS OF INCIDENT POLARIZATION UNIT VECTOR PARALLEL TO PLANE OF INCIDENCE |
| UIPRX<br>UIPRY<br>UIPRZ | X,Y,Z COMPONENTS OF INC/REFL POLARIZATION UNIT VECTOR PERPENDICULAR TO PLANE OF INCIDENCE |
| URPPX<br>URPPY<br>URPPZ | X,Y,Z COMPONENTS OF REFLECTED POLARIZATION UNIT VECTOR PARALLEL TO PLANE OF INCIDENCE |
| VT | X,Y,Z COMPONENTS OF POLARIZATION UNIT VECTOR TANGENT TO PLATE AND NORMAL TO RAY INCIDENT ON PLATE |
| VXS | MATRIX DEFINING SOURCE COORDINATE SYS AXES IN RCS COMPONENTS |
| XR | X,Y,Z COMPONENTS OF REFLECTION POINT LOCATION ON CYL |
| XRS | REFLECTION POINT ON PLATE (ALSO CYL REFL. POINT IMAGE LOCATION IN PLATE) ALSO CYLINDER REFLECTION POINT |

# CODE LISTING

```
C----------------------------------------------------------------------
      SUBROUTINE RCLRPL(ETH,EPH,MP)
C!!!
C!!!  COMPUTES THE G.O. FIELD REFLECTED FROM THE ELLIPTIC CYLINDER
C!!!  THEN REFLECTED FROM PLATE #MP
C!!!
      DIMENSION UN(2),UB(2),DI(3),DJ(3),XRS(3),VT(3)
      COMPLEX ETH,EPH,EX,EY,EZ,PH,EIPR,EIPP,ERX,ERY,ERZ,ERPR,ERPP
      COMPLEX EF,EG,A1,A2,ESTH,ESPH,EHTH,EHPH,TRAN
      LOGICAL LHIT,LRFS,LDEBUG,LTEST,LTRFJ
      COMMON/FUDGJ/TRAN,ESTH,ESPH,EHTH,EHPH,XR(3),RG,RHO1,SMAG,LTRFJ
      COMMON/GEOMEL/A,B,ZC(2),SNC(2),CNC(2),CTC(2)
      COMMON/SORINF/XS(3),VXS(3,3)
      COMMON/GEOPLA/X(14,6,3),V(14,6,3),VR(14,6,3),VN(14,3)
     2,MEP(14),MPX
      COMMON/PIS/PI,TPI,DPR,RPD
      COMMON/DIR/D(3),THSR,PHSR,SPS,CPS,STHS,CTHS
      COMMON/THPHUV/DT(3),DP(2)
      COMMON/BNDSCL/DTS,VTS(2),BTS(4)
      COMMON/TEST/LDEBUG,LTEST
      COMMON/CLRFS/LRFS(14)
      IF(LDEBUG) WRITE(6,900)
900   FORMAT(/' DEBUGGING RCLRPL SUBROUTINE')
      LTRFJ=.FALSE.
      IF(DTS.LT.-1.5) GO TO 12
C!!!  COMPUTE DIRECTION OF RAY INCIDENT ON PLATE
      CALL REFBP(PHJR,THJR,PHSR,THSR,MP)
      SPHJ=SIN(PHJR)
      CPHJ=COS(PHJR)
      STHJ=SIN(THJR)
      CTHJ=COS(THJR)
      DJ(1)=CPHJ*STHJ
      DJ(2)=SPHJ*STHJ
      DJ(3)=CTHJ
      DXY=XS(1)*CPHJ+XS(2)*SPHJ
      IF(DXY.GT.0.) GO TO 10
      DD1=BTS(1)*CPHJ+BTS(2)*SPHJ
      DD2=BTS(3)*CPHJ+BTS(4)*SPHJ
C!!!  CAN CYLINDER REFLECTION OCCUR?
      IF(DD1.GT.DTS.AND.DD2.GT.DTS) GO TO 12
10    CONTINUE
C!!!  COMPUTE CYLINDER REFLECTION POINT LOCATION
      CALL RFPTCL(PHJR,-MP,VR,DOTP,DD,S,LRFS(MP))
      IF(LDEBUG) WRITE(6,*) VR,DOTP,DD,S,LRFS(MP)
      IF(DOTP.LE.0.) GO TO 11
      XR(1)=A*COS(VR)
      XR(2)=B*SIN(VR)
      XR(3)=XS(3)+S*CTHJ/STHJ
C!!!  IS REFLECTION POINT ON CYLINDER?
      IF(XR(3).GT.ZC(1)+XR(1)*CTC(1).OR.
     2XR(3).LT.ZC(2)+XR(1)*CTC(2)) GO TO 11
      DO 15 N=1,3
15    XRS(N)=XR(N)
C!!!  DOES REFLECTION FROM PLATE OCCUR?
      CALL PLAINT(XRS,DJ,DHJT,-MP,LHIT)
      IF(.NOT.LHIT) GO TO 11
C!!!  IS RAY SHADOWED ANYWHERE?
      CALL PLAINT(XRS,D,DHT,MP,LHIT)
      IF(LHIT) GO TO 11
      CALL CYLINT(XRS,D,PHSR,DHT,LHIT,.TRUE.)
      IF(LHIT) GO TO 11
      CALL PLAINT(XR,DJ,DHT,MP,LHIT)
      IF(LHIT.AND.(DHT.LT.DHJT)) GO TO 11
      SXN=XS(1)-XR(1)
      SYN=XS(2)-XR(2)
      SZN=-S*CTHJ/STHJ
      SMAG=SQRT(SXN*SXN+SYN*SYN+SZN*SZN)
      SXN=SXN/SMAG
      SYN=SYN/SMAG
      SZN=SZN/SMAG
      PHIR=BTAN2(-SYN,-SXN)
      THIR=BTAN2(SQRT(SXN*SXN+SYN*SYN),-SZN)
      DI(1)=COS(PHIR)*SIN(THIR)
      DI(2)=SIN(PHIR)*SIN(THIR)
      DI(3)=COS(THIR)
      CALL PLAINT(XS,DI,DHIT,0,LHIT)
      IF(LHIT.AND.(DHIT.LT.SMAG)) GO TO 11
C!!!  COMPUTE SOURCE PATTERN FACTOR
      CALL SOURCE(EF,EG,EX,EY,EZ,THIR,PHIR,VXS)
      IF(LDEBUG) WRITE(6,*) EF,EG
      RG=DD*DD*DD/A/B
      CALL NANDB(UN,UB,VR)
      CTHW=UN(1)*DJ(1)+UN(2)*DJ(2)
      WR=BTAN2(SXN*UB(1)+SYN*UB(2),SZN)
      SW=SIN(WR)
      CW=COS(WR)
      SST2=SW*SW+CW*CW*CTHW*CTHW
      RHO2=SMAG
      RHO1=SMAG*RG*CTHW/(RG*CTHW+2.*SMAG*SST2)
      IF(LDEBUG) WRITE(6,*) RG,RHO1,RHO2,CTHI,SST2
C!!!  COMPUTE POLARIZATION UNIT VECTORS FOR
C!!!  INCIDENT AND REFLECTED FIELD AT CYLINDER
      UIPRX=SIN(WR-PI/2.)*UB(1)
      UIPRY=SIN(WR-PI/2.)*UB(2)
      UIPRZ=COS(WR-PI/2.)
      UIPPX=SYN*UIPRZ-SZN*UIPRY
      UIPPY=SZN*UIPRX-SXN*UIPRZ
      UIPPZ=SXN*UIPRY-SYN*UIPRX
      URPPX=UIPRY*DJ(3)-UIPRZ*DJ(2)
      URPPY=UIPRZ*DJ(1)-UIPRX*DJ(3)
      URPPZ=UIPRX*DJ(2)-UIPRY*DJ(1)
      PH=CEXP(CMPLX(0.,-TPI*SMAG))/SMAG
C!!!  COMPUTE FIELD COMPONENTS OF RAY INCIDENT ON CYL.
      EIPR=(UIPRX*EX+UIPRY*EY+UIPRZ*EZ)
      EIPP=(UIPPX*EX+UIPPY*EY+UIPPZ*EZ)
C!!!  COMPUTE LOCATION OF CYLINDER REFL. POINT
C!!!  IMAGE IN PLATE MP
      CALL IMAGE(XRS,XR,ANR,MP)
      GAM=XRS(1)*D(1)+XRS(2)*D(2)+XRS(3)*D(3)
      PH=PH*CEXP(CMPLX(0.,TPI*GAM))
      SQRH=SQRT(RHO1*RHO2)
C!!!  COMPUTE COMPONENTS OF CYLINDER REFL. FIELD
C!!!  PARALLEL AND PERPENDICULAR TO PLANE OF INC
      ERPR=-SQRH*PH*EIPR
      ERPP=SQRH*PH*EIPP
      TRAN=SQRH*PH
      ERX=ERPR*UIPRX+ERPP*URPPX
      ERY=ERPR*UIPRY+ERPP*URPPY
      ERZ=ERPR*UIPRZ+ERPP*URPPZ
C!!!  COMPUTE THETA AND PHI COMPONENTS OF CYLINDER
C!!!  REFLECTED FIELD
      EF=ERX*CPHJ*CTHJ+ERY*SPHJ*CTHJ-ERZ*STHJ
      EG=-ERX*SPHJ+ERY*CPHJ
C!!!  CALCULATE POLARIZATION VECTORS AND DOT PRODUCTS
C!!!  NECESSARY TO COMPUTE PLATE REFLECTED FIELD
      VT(1)=VN(MP,2)*D(3)-VN(MP,3)*D(2)
      VT(2)=VN(MP,3)*D(1)-VN(MP,1)*D(3)
      VT(3)=VN(MP,1)*D(2)-VN(MP,2)*D(1)
      C11=VN(MP,1)*CPHJ*CTHJ+VN(MP,2)*SPHJ*CTHJ-VN(MP,3)*STHJ
      C12=-VN(MP,1)*SPHJ+VN(MP,2)*CPHJ
      C21=VT(1)*CPHJ*CTHJ+VT(2)*SPHJ*CTHJ-VT(3)*STHJ
      C22=-VT(1)*SPHJ+VT(2)*CPHJ
```

```
C!!!   COMPUTE FIELD REFLECTED FROM PLATE
       A1=EF*C11+EG*C12
       A2=EF*C21+EG*C22
       C11=VN(MP,1)*DT(1)+VN(MP,2)*DT(2)+VN(MP,3)*DT(3)
       C12=VN(MP,1)*DP(1)+VN(MP,2)*DP(2)
       C21=VT(1)*DT(1)+VT(2)*DT(2)+VT(3)*DT(3)
       C22=VT(1)*DP(1)+VT(2)*DP(2)
       A3=C11*C22-C12*C21
C!!!   COMPUTE THETA AND PHI REFLECTED FIELD COMPONENTS
       ETH=(A1*C22+A2*C12)/A3
       EPH=-(A2*C11+A1*C21)/A3
C!!!   COMPUTE THETA AND PHI COMPONENTS OF HARD AND
C!!!   SOFT COMPONENTS OF RAY INCIDENT ON CYLINDER
       ERX=EIPR*UIPRX
       ERY=EIPR*UIPRY
       ERZ=EIPR*UIPRZ
       ESTH=ERX*CPHJ*CTHJ+ERY*SPHJ*CTHJ-ERZ*STHJ
       ESPH=-ERX*SPHJ+ERY*CPHJ
       ERX=EIPP*URPPX
       ERY=EIPP*URPPY
       ERZ=EIPP*URPPZ
       EHTH=ERX*CPHJ*CTHJ+ERY*SPHJ*CTHJ-ERZ*STHJ
       EHPH=-ERX*SPHJ+ERY*CPHJ
       GO TO 905
 12    LRFS(MP)=.FALSE.
 11    LTRFJ=.TRUE.
       ETH=(0.,0.)
       EPH=(0.,0.)
 905   CONTINUE
       IF(.NOT.LTEST) RETURN
       WRITE(6,901)
 901   FORMAT(/,' TESTING RCLRPL SUBROUTINE')
       WRITE(6,*) ETH,EPH,MP
       RETURN
       END
```

## REFBP

### PURPOSE

To calculate incident ray direct:on needed in order to obtain reflected ray in a given direction off of a spe:ified plate.

### PERTINENT GEOMETRY

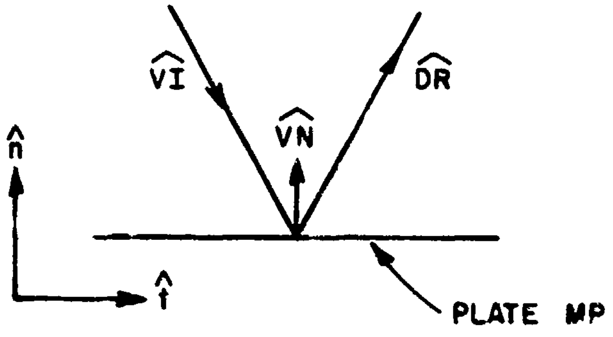


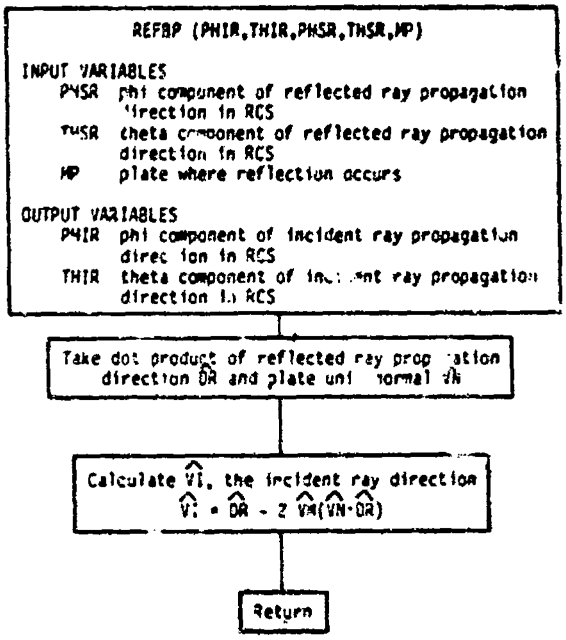
Figure 88--Illustration of incident and reflected rays on plate.

### METHOD

$\hat{VI}$ is found by imaging $\hat{DR}$ into plate MP:

$\hat{VI} = \hat{DR} - 2(\hat{VN} \cdot \hat{DR})\ \hat{VN}$.

## FLOW DIAGRAM


REFBP (PHIR,THIR,PHSR,THSR,NP)
INPUT VARIABLES
PHSR phi component of reflected ray propagation direction in RCS
THSR theta component of reflected ray propagation direction in RCS
NP plate where reflection occurs
OUTPUT VARIABLES
PHIR phi component of incident ray propagation direction in RCS
THIR theta component of incident ray propagation direction in RCS
Take dot product of reflected ray propagation direction $\hat{DR}$ and plate unit normal $\hat{VN}$
Calculate $\widehat{VI}$, the incident ray direction
$\widehat{VI} = \widehat{DR} - 2\,\widehat{VN}(\widehat{VN}\cdot\widehat{DR})$
Return

## SYMBOL DICTIONARY

| | |
|---|---|
| CPS | COSINE OF PHSR |
| CTS | COSINE OF THSR |
| DN | CROSS PRODUCT OF DR AND VN |
| DR | REFLECTED RAY PROPAGATION DIRECTION IN X,Y,Z RCS COMPONENTS |
| END | ERROR DETECTION VARIABLE |
| NP | PLATE UPON WHICH REFLECTION OCCURS |
| PHIR | PHI COMPONENT OF INCIDENT RAY PROPAGATION DIRECTION IN RCS |
| PHSR | PHI COMPONENT OF REFLECTED RAY PROPAGATION DIRECTION IN RCS |
| SPS | SINE OF PHSR |
| STS | SINE OF THSR |
| THIR | THETA COMPONENT OF INCIDENT RAY PROPAGATION DIRECTION IN RCS |
| THSR | THETA COMPONENT OF REFLECTED RAY PROPAGATION DIRECTION IN RCS |
| VI | X,Y,Z COMPONENTS OF INCIDENT RAY PROPAGATION DIRECTION IN RCS |
| VIN | DOT PRODUCT OF PLATE NORMAL AND VI |

## CODE LISTING

```
C---------------------------------------------------------------------
      SUBROUTINE REFBP(PHIR,THIR,PHSR,THSR,MP)
C!!!
C!!!  DETERMINE INCIDENT RAY DIRECTION (PHIR,THIR)
C!!!  IF RAY REFLECTED FROM PLATE #MP IS IN (PHSR,THSR) DIRECTION.
C!!!
      DIMENSION DR(3),VI(3)
      COMMON/GEOPLA/X(14,6,3),V(14,6,3),VP(14,6,3),VN(14,3)
     2,NEP(14),MPX
      COMMON/PIS/PI,TPI,DPR,RPD
      CPS=COS(PHSR)
      SPS=SIN(PHSR)
      CTS=COS(THSR)
      STS=SIN(THSR)
      DR(1)=CPS*STS
      DR(2)=SPS*STS
      DR(3)=CTS
C!!!  TAKE DOT PRODUCT OF DR AND VN
      DN=VN(MP,1)*DR(1)+VN(MP,2)*DR(2)+VN(MP,3)*DR(3)
C!!!  CALCULATE VI, THE INC RAY DIRECTION
      DO 10 N=1,3
 10   VI(N)=DR(N)-2.*DN*VN(MP,N)
C!!!  CONVERT VI TO SPHERICAL ANGLES IN RCS
      PHIR=ATAN2(VI(2),VI(1))
      THIR=ATAN2(SQRT(VI(1)*VI(1)+VI(2)*VI(2)),VI(3))
      VIN=VN(MP,1)*VI(1)+VN(MP,2)*VI(2)+VN(MP,3)*VI(3)
      ERD=ABS(DN+VIN)
      IF(ERD.GT.1.E-5) WRITE(6,1) ERD,PHSR,THSR
 1    FORMAT(' ERROR IN REFBP= ',3F12.5)
      RETURN
      END
```

## REFCAP

### PURPOSE

To calculate the far-zone electric field resulting from the reflection of the source off of a given cylinder end cap.

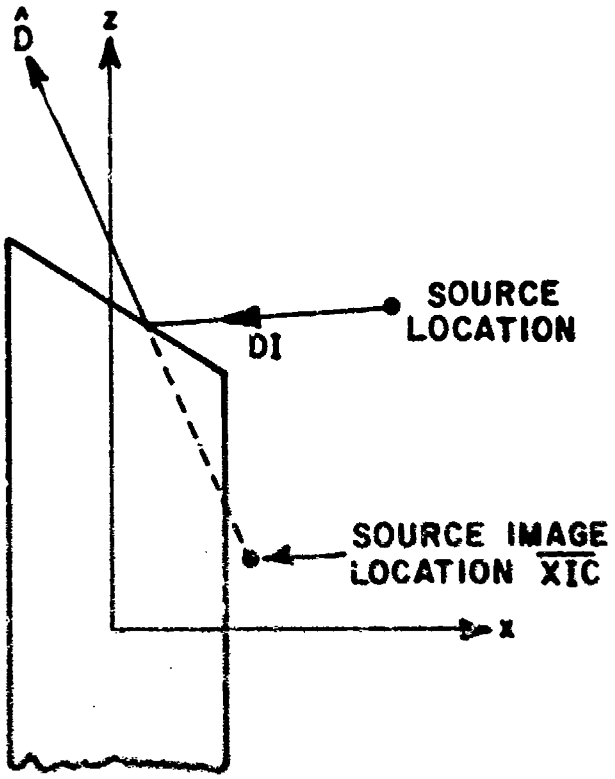


Figure 89--Illustration of source ray reflection from end cap.

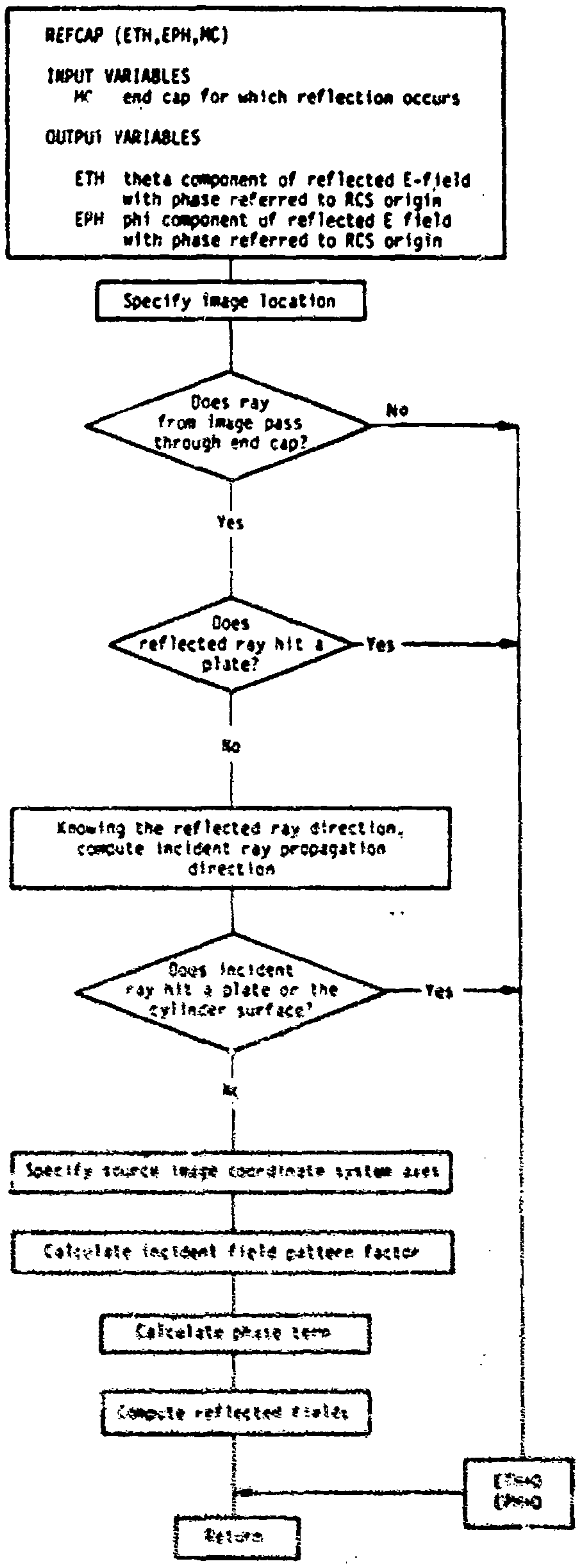
### METHOD

The field reflected from a cylinder end cap is found using image theory. First the ray path is checked to insure that the reflection point is on the end cap and that the ray is not shadowed. The fields are then calculated using the SOURCE subroutine with the source coordinates oriented from image theory so that the proper boundary conditions are met at the surface of the end cap. The phase is referred to the reference coordinate system origin using the factor $e^{jk\hat{D}\cdot\overline{XIC}}$. The reflected field has the form

$$\overline{E}^r(r,\theta,\phi) = W_m(ETH\hat{\theta}+EPH\hat{\phi})\,\frac{e^{-jkR}}{R} \; .$$

The factor $\frac{e^{-jkR}}{R}$ and source weight ($W_m$) are added elsewhere in the code.

## FLOW DIAGARM


REFCAP (ETH,EPH,MC)
INPUT VARIABLES
MC end cap for which reflection occurs
OUTPUT VARIABLES
ETH theta component of reflected E-field with phase referred to RCS origin
EPH phi component of reflected E field with phase referred to RCS origin
Specify image location
Does ray from image pass through end cap?
No
Yes
Does reflected ray hit a plate?
Yes
No
Knowing the reflected ray direction, compute incident ray propagation direction
Does incident ray hit a plate or the cylinder surface?
Yes
No
Specify source image coordinate system axes
Calculate incident field pattern factor
Calculate phase term
Compute reflected fields
ETH=0
EPH=0
Return

SYMBOL DICTIONARY

| | |
|---|---|
| DHT | DISTANCE FROM SOURCE TO HIT POINT ON END CAP (FROM CAPINT) |
| DHIT | DISTANCE FROM SOURCE TO HIT POINT ON PLATE (FROM PLAINT) |
| DI | UNIT VECTOR OF INCIDENT RAY PROPAGATION DIRECTION |
| DN | DOT PRODUCT OF REFLECTED RAY PROP DIR AND END CAP UNIT NORMAL |
| DNI | DOT PRODUCT OF INCIDENT RAY AND END CAP UNIT NORMAL |
| EF | PATTERN FACTOR FOR THETA COMPONENT OF INCIDENT E FIELD |
| EG | PATTERN FACTOR FOR PHI COMPONENT OF INCIDENT E FIELD |
| EPH | PHI COMPONENT OF REFLECTED E FIELD IN RCS |
| ETH | THETA COMPONENT OF REFLECTED E FIELD IN RCS |
| EX | PHASE TERM |
| GAM | PHASE TERM PARAMETER |
| LHIT | SET TRUE IF RAY HITS PLATE(FROM PLAINT) |
| MC | END CAP WHERE REFLECTION OCCURS |
| N | DO LOOP VARIABLE |
| NC | SIGN CHANGE VARIABLE |
| NI | DO LOOP VARIABLE |
| NJ | DO LOOP VARIABLE |
| VAX | X,Y,Z COMPONENTS DEFINING THE IMAGE SOURCE COORDINATE SYSTEM IN (XYZ) RCS COMPONENTS |
| VN | UNIT NORMAL TO END CAP IN RCS (X,Y,Z) COMPONENTS |
| XIS | SOURCE IMAGE LOCATION |

## CODE LISTING

```
C------------------------------------------------------------------------
      SUBROUTINE REFCAP(ETH,EPH,MC)
C!!!
C!!!  COMPUTES THE REFLECTED FIELD FROM THE END CAPS
C!!!  OF THE ELLIPTIC CYLINDER
C!!!
      COMPLEX ETH,EPH,EF,EG,EIX,EIY,EIZ,EX
      DIMENSION XIS(3),DI(3),VN(3),VAX(3,3)
      LOGICAL LHIT,LDEBUG,LTEST
      COMMON/DIR/D(3),THSR,PHSR,SPHS,CPHS,STHS,CTHS
      COMMON/SORINF/XS(3),VXS(3,3)
      COMMON/IMCINF/XIC(2,3),VXIC(3,3,2)
      COMMON/GEOMEL/A,B,ZC(2),SNC(2),CNC(2),CTC(2)
      COMMON/PIS/PI,TPI,DPR,RPD
      COMMON/TEST/LDEBUG,LTEST
      IF(LDEBUG) WRITE(6,9ØØ)
9ØØ   FORMAT(/,' DEBUGGING REFCAP SUBROUTINE')
C!!!  SPECIFY IMAGE LOCATION
      DO 5 N=1,3
5     XIS(N)=XIC(MC,N)
C!!!  DOES RAY FROM IMAGE PASS THRU DISK
      CALL CAPINT(XIS,D,DHT,MC,LHIT)
      IF(.NOT.LHIT) GO TO 3Ø
C!!!  DOES REFL. RAY HIT A PLATE
      CALL PLAINT(XIS,D,DHIT,Ø,LHIT)
      IF(LHIT) GO TO 3Ø
C!!!  KNOWING OBS. DIR. COMPUTE THE INCIDENT RAY PROPAGATION
C!!!  DIRECTION
      NC=MC
      IF(MC.GT.1) NC=-1
      VN(1)=-NC*CNC(MC)
      VN(2)=Ø.
      VN(3)=NC*SNC(MC)
      DN=VN(1)*D(1)+VN(2)*D(2)+VN(3)*D(3)
      DO 1Ø N=1,3
1Ø    DI(N)=D(N)-2.*DN*VN(N)
C!!!  DOES RAY FROM SOURCE HIT A PLATE
      CALL PLAINT(XS,DI,DHIT,Ø,LHIT)
      IF(LHIT.AND.(DHIT.LT.DHT)) GO TO 3Ø
C!!!  DOES RAY FROM SOURCE HIT THE CYLINDER
      DNI=VN(1)*DI(1)+VN(2)*DI(2)+VN(3)*DI(3)
      IF(DNI.GE.Ø.) GO TO 3Ø
C!!!  SPECIFY SOURCE IMAGE AXES
      DO 2Ø NJ=1,3
      DO 2Ø NI=1,3
2Ø    VAX(NI,NJ)=VXIC(NI,NJ,MC)
C!!!  CALCULATE INCIDENT FIELD PATTERN FACTOR
      CALL SOURCE(EF,EG,EIX,EIY,EIZ,THSR,PHSR,VAX)
      IF(LDEBUG) WRITE(6,*) XIS
      IF(LDEBUG) WRITE(6,*) EF,EG
C!!!  CALCULATE PHASE TERM (REFER PHASE TO RCS ORIGIN)
      GAM=XIC(MC,1)*D(1)+XIC(MC,2)*D(2)+XIC(MC,3)*D(3)
      EX=CEXP(CMPLX(Ø.,TPI*GAM))
      ETH=EF*EX
      EPH=EG*EX
      RETURN
3Ø    CONTINUE
      ETH=(Ø.,Ø.)
      EPH=(Ø.,Ø.)
      IF(.NOT.LTEST) RETURN
      WRITE(6,91Ø)
91Ø   FORMAT(/,' TESTING REFCAP SUBROUTINE')
      WRITE(6,*) ETH,EPH,MC
      RETURN
      END
```

## REFCYL

PURPOSE

To calculate the geometrical optics field due to reflection of the source field off of the cylinder surface and generate data used in subroutine SCTCYL.

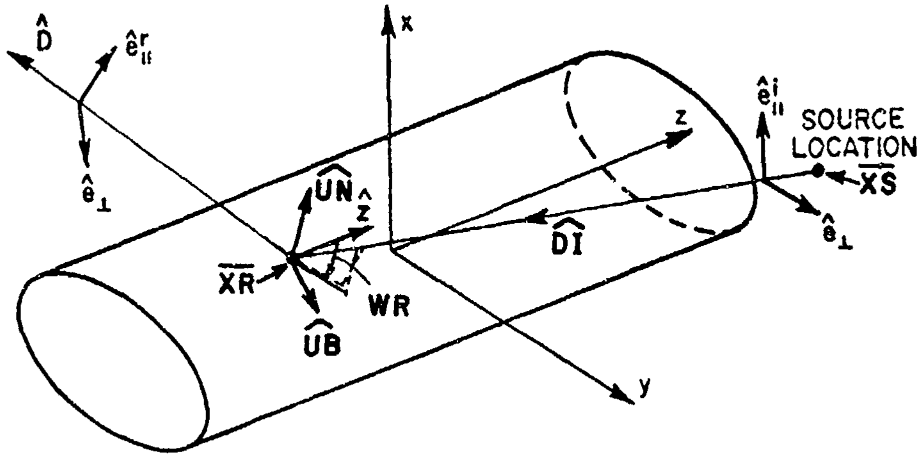


Figure 90 -- Geometry of ray reflected from cylinder.

$\hat{e}_\perp$ = UIPRX $\hat{x}$ + UIPRY $\hat{y}$ + UIPRZ $\hat{z}$

$\hat{e}^i_\parallel$ = UIPPX $\hat{x}$ + UIPPY $\hat{y}$ + UIPPZ $\hat{z}$

$\hat{e}^r_\parallel$ = URPPX $\hat{x}$ + URPPY $\hat{y}$ + URPPZ $\hat{z}$

$\widehat{UN}$ = UN(1)$\hat{x}$ + UN(2)$\hat{y}$ = normal to cylinder

$\widehat{UB}$ = UB(1)$\hat{x}$ + UB(2)$\hat{y}$ = tangent to cylinder

$\overline{XR}$ = reflection point = $\hat{x}$ XR(1) + $\hat{y}$ XR(2) + $\hat{z}$ XR(3)

$\overline{XS}$ = $\hat{x}$ XS(1) + $\hat{y}$ XS(2) + $\hat{z}$ XS(3)

## METHOD

Subroutine REFCYL functions as a service routine for subroutine SCTCYL, where the actual cylinder fields are computed. The geometrical optics reflected field components ETH and EPH computed in REFCYL are used only for reference purposes (when LOUT is set true). The field components calculated in REFCYL which are used in SCTCYL are the hard and soft components of the source field incident on the cylinder at the reflection point. These components, along with several other useful parameters are passed to subroutine SCTCYL through common block FUDG.

The geometrical optics fields [4] in the far field have the form

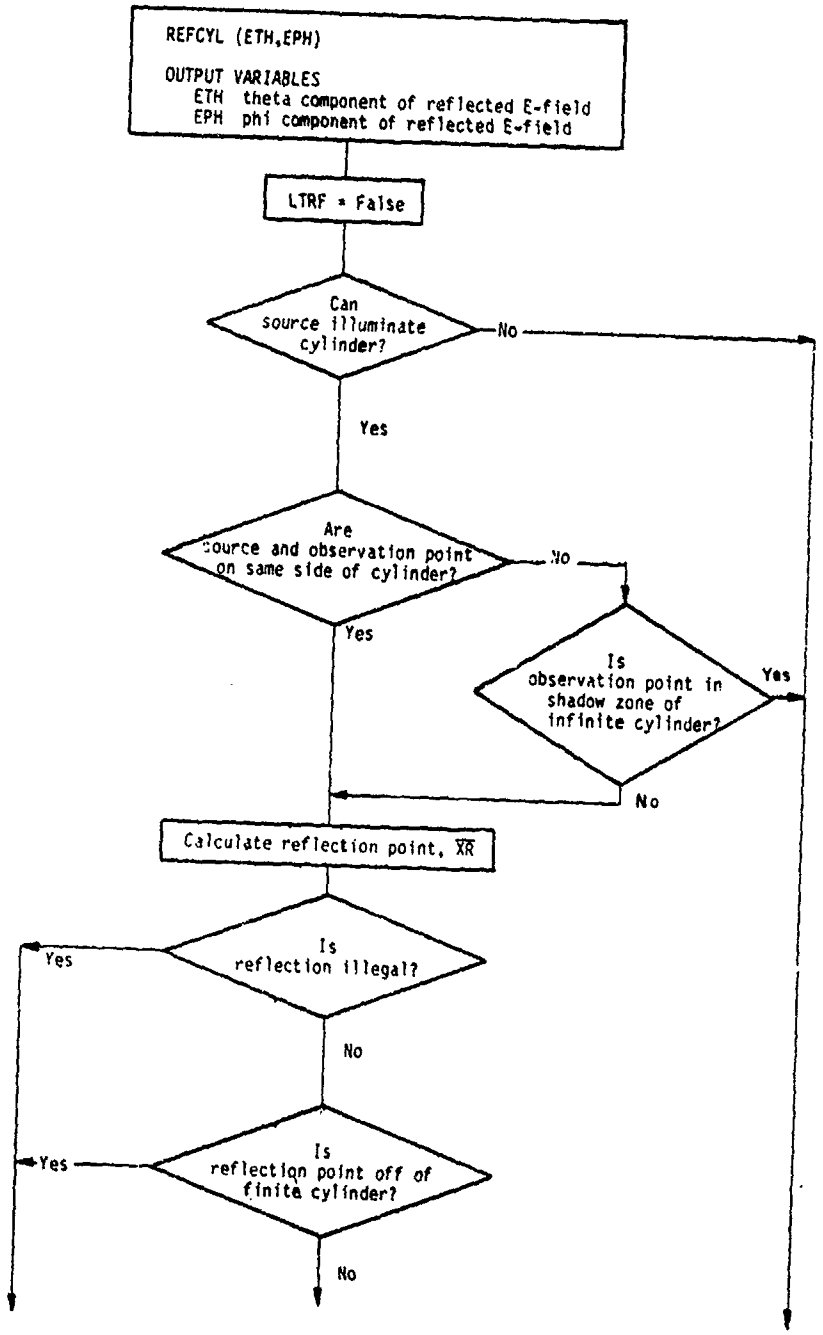
$$\bar{E}^r = \bar{E}^i(Q_R) \cdot \bar{\bar{R}}\sqrt{\rho_1^r\rho_2^r}\,\frac{e^{-jks}}{s}$$

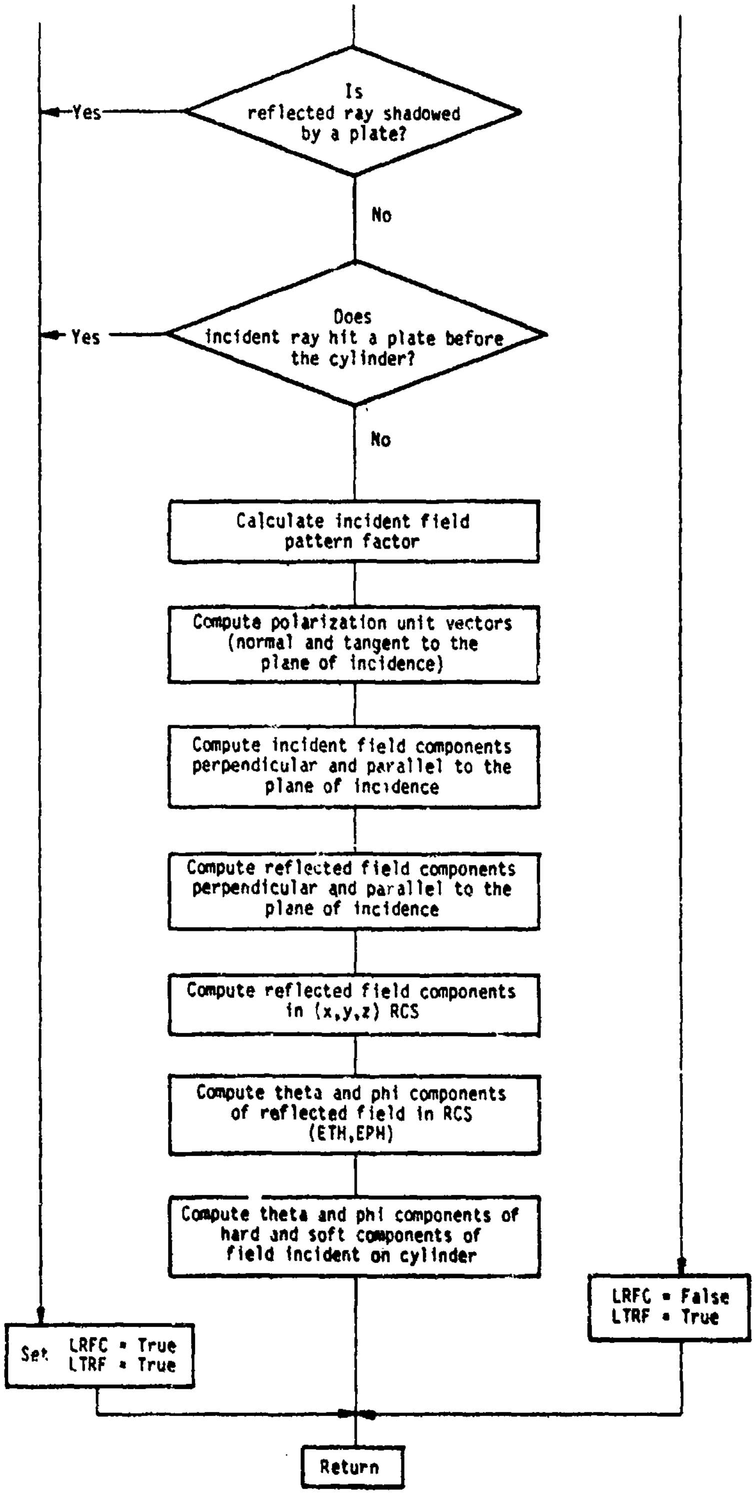
where $\bar{E}^i(Q_R)$ is the incident field at the reflection point, $\bar{\bar{R}}$ is the dyadic reflection coefficient, s is the distance from the reflection point to the far field, and $\sqrt{\rho_1^r\rho_2^r}/s$ is the far-field spread factor for the field. The caustic distances $\rho_1^r$ and $\rho_2^r$ and further details to the solution are given on pages 105-107 of Reference 1 . The phase of the reflected field is referred to the reference coordinate system origin so that $\frac{e^{-jks}}{s} = e^{jk\hat{D}\cdot\overline{XR}}\,\frac{e^{-jkR}}{R}$ . The reflected field then has the form

$$\bar{E}^r = W_m(ETH\hat{\theta} + EPH\hat{\phi})\,\frac{e^{-jkR}}{R}$$

where the factor $\frac{e^{-jkR}}{R}$ and the source weight ($W_m$) are added elsewhere in the code.

FLOW DIAGRAM


REFCYL (ETH,EPH)
OUTPUT VARIABLES
ETH theta component of reflected E-field
EPH phi component of reflected E-field
LTRF = False
Can source illuminate cylinder?
No
Yes
Are source and observation point on same side of cylinder?
No
Yes
Is observation point in shadow zone of infinite cylinder?
Yes
No
Calculate reflection point, $\overline{XR}$
Is reflection illegal?
Yes
No
Is reflection point off of finite cylinder?
Yes
No


Is reflected ray shadowed by a plate?
Yes
No
Does incident ray hit a plate before the cylinder?
Yes
No
Calculate incident field pattern factor
Compute polarization unit vectors (normal and tangent to the plane of incidence)
Compute incident field components perpendicular and parallel to the plane of incidence
Compute reflected field components perpendicular and parallel to the plane of incidence
Compute reflected field components in (x,y,z) RCS
Compute theta and phi components of reflected field in RCS (ETH,EPH)
Compute theta and phi components of hard and soft components of field incident on cylinder
Set LRFC = True LTRF = True
LRFC = False LTRF = True
Return

## SYMBOL DICTIONARY

| | |
|---|---|
| CTHI | DOT PRODUCT OF CYLINDER NORMAL AND REFL PROP DIR UNIT VECTOR |
| CW | COSINE OF WR |
| D | PROPAGATION DIRECTION AFTER REFL. IN (X,Y,Z) RCS COMPONENTS |
| D12 | DOT PRODUCT OF SOURCE VECTORS TANGENT TO CYLINDER (2-D) |
| DD | NORMALIZATION CONSTANT FOR REFL. PT. UNIT NORMAL (FROM RFPTCL) |
| DD1 | DOT PRODUCT OF UNIT VECTOR OF PROPAGATION DIRECTION AND CYLINDER TANGENT UNIT VECTOR THROUGH TAN POINT 1 (2-D) |
| DD2 | DOT PRODUCT OF UNIT VECTOR OF PROPAGATION DIRECTION AND CYLINDER TANGENT UNIT VECTOR THROUGH TAN POINT 2 (2-D) |
| DHIT | DISTANCE FROM SOURCE TO HIT POINT (FROM PLAINT) |
| DI | X,Y, AND Z COMPONENTS OF INCIDENT RAY DIRECTION IN RCS |
| DOTP | DIFFERENCE OF DOT PRODUCTS RETURNED FROM SUB RFPTCL (2-D) |
| DXY | DOT PRODUCT OF VECTOR FROM ORIGIN TO SOURCE AND PROP. DIR (2-D) |
| EF | PATTERN FACTOR OF THETA COMPONENT OF INCIDENT FIELD IN RCS |
| EG | PATTERN FACTOR OF PHI COMPONENT OF INCIDENT FIELD IN RCS |
| EHPH | PHI COMPONENT OF THE HARD COMPONENT OF FIELD INC ON CYL |
| EHTH | THETA COMPONENT OF THE HARD COMPONENT OF FIELD INC ON CYL |
| EIPP | INCIDENT FIELD COMPONENT PARALLEL TO PLANE OF INCIDENCE |
| EIPR | INCIDENT FIELD COMPONENT PERPENDICULAR TO PLANE OF INC |
| EPH | PHI COMPONENT OF REFLECTED E-FIELD |
| ERPP | REFLECTED FIELD COMPONENT PARALLEL TO PLANE OF INCIDENCE |
| ERPR | REFLECTED FIELD COMPONENT PERPENDICULAR TO PLANE OF INC. |
| ERX<br>ERY<br>ERZ | X,Y,Z COMPONENTS OF REFLECTED FIELD IN RCS (ALSO USED TO DEFINE COMPONENTS INCIDENT ON CYLINDER) |
| ESPH | PHI COMPONENT OF THE SOFT COMPONENT OF FIELD INC ON CYL |
| ESTH | THETA COMPONENT OF THE SOFT COMPONENT OF FIELD INC ON CYL |
| ETH | THETA COMPONENT OF REFLECTED E FIELD |
| EX<br>EY<br>EZ | PATTERN FACTOR OF X,Y,Z COMPONENTS OF INCIDENT FIELD IN RCS |
| LHIT | SET TRUE IF RAY HITS PLATE (FROM PLAINT) |
| LRFC | SET TRUE IF REFL DATA IS AVAILABLE FROM PREVIOUS PATTERN ANGLE (OR FOR NEXT PATTERN ANGLE WHEN LEAVING ROUTINE) |
| LTRF | SET TRUE IF G.O. REFLECTED FIELD DOES NOT EXIST |
| PH | PHASE AND MAGNITUDE CONSTANT FOR INCIDENT OR REFLECTED FIELD |
| PHIR | PHI COMPONENT OF INCIDENT RAY DIRECTION |
| RG | PARAMETER USED IN TRANSITION FUNCTION |
| RHO1 | RAY SPREADING RADIUS IN PLANE OF CYL CURVATURE AT REFL. PT. |
| RHO2 | RAY SPREADING RADIUS IN PLANE NORMAL TO PLANE OF INCIDENCE AT REFLECTION POINT |
| S | DISTANCE FROM SOURCE TO REFL. POINT IN X-Y PLANE |
| SMAG | DISTANCE FROM SOURCE TO REFLECTION POINT |
| SQRH | SPREADING FACTOR |
| SW | SINE OF WR |
| SXN<br>SYN<br>SZN | X,Y, AND Z COMPONENTS OF UNIT VECTOR OF RAY FROM REFL. POINT TO SOURCE IN RCS |
| THIR | THETA COMPONENT OF INCIDENT RAY DIRECTION |
| TRAN | PARAMETER USED IN TRANSITION FUNCTION |
| TX1 | X COMPONENT OF SOURCE VECTOR TANGENT TO TAN POINT 1 (2-D) |
| TX2 | X COMPONENT OF SOURCE VECTOR TANGENT TO TAN POINT 2 (2-D) |
| TY1 | Y COMPONENT OF SOURCE VECTOR TANGENT TO TAN POINT 1 (2-D) |
| TY2 | Y COMPONENT OF SOURCE VECTOR TANGENT TO TAN POINT 2 (2-D) |
| UB | X,Y COMPONENTS OF UNIT VECTOR TANGENT TO CYLINDER REFLECTION POINT IN RCS (2-D) |
| UIPPX<br>UIPPY<br>UIPPZ | X,Y,Z COMPONENTS OF INCIDENT FIELD POLARIZATION UNIT VECTOR PARALLEL TO PLANE OF INCIDENCE |
| UIPRX<br>UIPRY<br>UIPRZ | X,Y,Z COMPONENTS OF INC/REFL FIELD POLARIZATION UNIT VECTOR PERPENDICULAR TO PLANE OF INCIDENCE |
| UN | X,Y COMPONENTS OF UNIT NORMAL TO CYLINDER REFL POINT IN RCS (2-D) |

| | |
|---|---|
| URFPX<br>URFPY<br>URPPZ | X,Y,Z COMPONENTS OF REFL FIELD POLARIZATION UNIT VECTOR PARALLEL TO PLANE OF INCIDENCE |
| VR | ELL. ANGLE DEFINING REFLECTION POINT IN ERCS. |
| VXS | X,Y,Z COMPONENTS OF UNIT VECTORS DEFINING SOURCE COORDINATE SYSTEM AXES IN RCS |
| WR | PHI ANGLE DEFINING PROPAGATION DIRECTION IN CYL REFL. POINT COORD SYSTEM |
| XR | LOCATION OF REFLECTION POINT IN (X,Y,Z) REF COORD SYS. |

## CODE LISTING

```
C-------------------------------------------------------------------
      SUBROUTINE REFCYL(ETH,EPH)
C!!!
C!!!  COMPUTES THE REFLECTED FIELD OF THE ELLIPTIC CYLINDER
C!!!
      DIMENSION UN(2),UB(2),DI(3)
      COMPLEX ETH,EPH,EX,EY,EZ,PH,EIPR,EIPP,ERX,ERY,ERZ,ERPR,ERPP
      COMPLEX ESTH,ESPH,EHTH,EHPH,TRAN,EF,EG
      LOGICAL LHIT,LRFC,LTRF,LDEBUG,LTEST
      COMMON/FUDG/TRAN,ESTH,ESPH,EHTH,EHPH,XR(3),RG,RHOI,SMAG,LTRF
      COMMON/GEOMEL/A,B,ZC(2),SNC(2),CNC(2),CTC(2)
      COMMON/SORINF/XS(3),VXS(3,3)
      COMMON/PIS/PI,TPI,DPR,RPD
      COMMON/DIR/D(3),THSR,PHSR,SPS,CPS,STHS,CTHS
      COMMON/THPHUV/DT(3),DP(2)
      COMMON/ENDSCL/DTS,VTS(2),BTS(4)
      COMMON/TEST/LDEBUG,LTEST
      COMMON/CLRFC/LRFC
      IF(LDEBUG) WRITE(6,900)
900   FORMAT(/,' DEBUGGING REFCYL SUBROUTINE')
      LTRF=.FALSE.
C!!!  CAN SOURCE ILLUMINATE CYLINDER?
      IF(DTS.LT.-1.5) GO TO 12
      DXY=XS(1)*CPS+XS(2)*SPS
C!!!  IS SOURCE AND OBSERVATION POINT ON SAME SIDE OF CYLINDER?
      IF(DXY.GT.0.) GO TO 10
      D12=DTS
      TX1=BTS(1)
      TY1=BTS(2)
      TX2=BTS(3)
      TY2=BTS(4)
      DD1=TX1*CPS+TY1*SPS
      DD2=TX2*CPS+TY2*SPS
C!!!  IS OBSERVATION POINT IN SHADOW ZONE OF INFINITE CYLINDER?
      IF(DD1.GT.D12.AND.DD2.GT.D12) GO TO 12
10    CONTINUE
C!!!  CALCULATE REFLECTION POINT
      CALL RFPTCL(PHSR,0.VR,DOTP,DD,S,LRFC)
      IF(LDEBUG) WRITE(6,*) VR,DOTP,DD,S,LRFC
C!!!  IS REFLECTION ILLEGAL?
      IF(DOTP.LE.0.)GO TO 11
      XR(1)=A*COS(VR)
      XR(2)=B*SIN(VR)
      XR(3)=XS(3)+S*CTHS/STHS
C!!!  IS REFLECTION POINT OFF OF FINITE CYLINDER?
      IF(XR(3).GT.ZC(1)+XR(1)*CTC(1).OR.
     2XR(3).LT.ZC(2)+XR(1)*CTC(2)) GO TO 11
C!!!  IS REFLECTED RAY SHADOWED BY A PLATE?
      CALL PLAINT(XR,D,DHIT,0,LHIT)
      IF(LHIT) GO TO 11
      SXN=XS(1)-XR(1)
      SYN=XS(2)-XR(2)
      SZN=-S*CTHS/STHS
      SMAG=SQRT(SXN*SXN+SYN*SYN+SZN*SZN)
      SXN=SXN/SMAG
      SYN=SYN/SMAG
      SZN=SZN/SMAG
      PHIR=ATAN2(-SYN,-SXN)
      THIR=ATAN2(SQRT(SXN*SXN+SYN*SYN),-SZN)
      DI(1)=COS(PHIR)*SIN(THIR)
      DI(2)=SIN(PHIR)*SIN(THIR)
      DI(3)=COS(THIR)
C!!!  DOES INCIDENT RAY HIT PLATE BEFORE CYLINDER?
      CALL PLAINT(XS,DI,DHIT,0,LHIT)
      IF(LHIT.AND.(DHIT.LT.SMAG)) GO TO 11
C!!!  CALCULATE INCIDENT FIELD PATTERN FACTORS
```

```
      CALL SOURCE(EF,EG,EX,EY,EZ,THIR,PHIR,VXS)
      IF(LDEBUG) WRITE(6,*) EF,EG
      RG=DD*DD*DD/A/B
      CALL NANDB(UN,UB,VR)
      CTHI=UN(1)*D(1)+UN(2)*D(2)
      WR=BTAN2(SXN*UB(1)+SYN*UB(2),SZN)
      SW=SIN(WR)
      CW=COS(WR)
      SST2=SW*SW+CW*CW*CTHI*CTHI
      RHO2=SMAG
      RHO1=SMAG*RG*CTHI/(RG*CTHI+2.*SMAG*SST2)
      IF(LDEBUG) WRITE(6,*) RG,RHO1,RHO2,CTHI,SST2
C!!!  COMPUTE FIELD POLARIZATION UNIT VECTORS (PERPENDICULAR
C!!!  AND PARALLEL TO PLANE OF INCIDENCE)
      UIPRX=SIN(WR-PI/2.)*UB(1)
      UIPRY=SIN(WR-PI/2.)*UB(2)
      UIPRZ=COS(WR-PI/2.)
      UIPPX=SYN*UIPRZ-SZN*UIPRY
      UIPPY=SZN*UIPRX-SXN*UIPRZ
      UIPPZ=SXN*UIPRY-SYN*UIPRX
      URPPX=UIPRY*D(3)-UIPRZ*D(2)
      URPPY=UIPRZ*D(1)-UIPRX*D(3)
      URPPZ=UIPRX*D(2)-UIPRY*D(1)
      PH=CEXP(CMPLX(0.,-TPI*SMAG))/SMAG
C!!!  COMPUTE INCIDENT FIELD COMPONENTS PERPENDICULAR AND
C!!!  PARALLEL TO PLANE OF INCIDENCE
      EIPR=(UIPRX*EX+UIPRY*EY+UIPRZ*EZ)
      EIPP=(UIPPX*EX+UIPPY*EY+UIPPZ*EZ)
      PH=PH*CEXP(CMPLX(0.,TPI*(XR(1)*D(1)+XR(2)*D(2)+XR(3)*D(3))))
      SQRH=SQRT(RHO1*RHO2)
C!!!  COMPUTE REFLECTED FIELD COMPONENTS PERPENDICULAR AND
C!!!  PARALLEL TO PLANE OF INCIDENCE
      ERPR=-SQRH*PH*EIPR
      ERPP=SQRH*PH*EIPP
      TRAN=SQRH*PH
C!!!  COMPUTE REFLECTED FIELD COMPONENTS IN (XYZ) RCS COMPONENTS
      ERX=ERPR*UIPRX+ERPP*URPPX
      ERY=ERPR*UIPRY+ERPP*URPPY
      ERZ=ERPR*UIPRZ+ERPP*URPPZ
C!!!  COMPUTE THETA AND PHI COMPONENTS OF REFLECTED FIELD IN RCS
      ETH=ERX*DT(1)+ERY*DT(2)+ERZ*DT(3)
      EPH=ERX*DP(1)+ERY*DP(2)
C!!!  COMPUTE THETA AND PHI COMPONENTS OF HARD AND SOFT
C!!!  COMPONENTS OF FIELD INCIDENT ON CYLINDER
      ERX=EIPR*UIPRX
      ERY=EIPR*UIPRY
      ERZ=EIPR*UIPRZ
      ESTH=ERX*DT(1)+ERY*DT(2)+ERZ*DT(3)
      ESPH=ERX*DP(1)+ERY*DP(2)
      ERX=EIPP*URPPX
      ERY=EIPP*URPPY
      ERZ=EIPP*URPPZ
      EHTH=ERX*DT(1)+ERY*DT(2)+ERZ*DT(3)
      EHPH=ERX*DP(1)+ERY*DP(2)
      GO TO 905
   42 LRFC=.FALSE.
   41 LIRF=.TRUE.
      ETH=(0.,0.)
      EPH=(0.,0.)
  905 CONTINUE
      IF(.NOT.LTEST) RETURN
      WRITE(6,910)
  910 FORMAT(/,' TESTING REFCYL SUBROUTINE')
      WRITE(6,*) ETH,EPH
      RETURN
      END
```

## REFPLA

### PURPOSE

To calculate the far-zone electric field due to single reflection off of a given plate.

### PERTINENT GEOMETRY

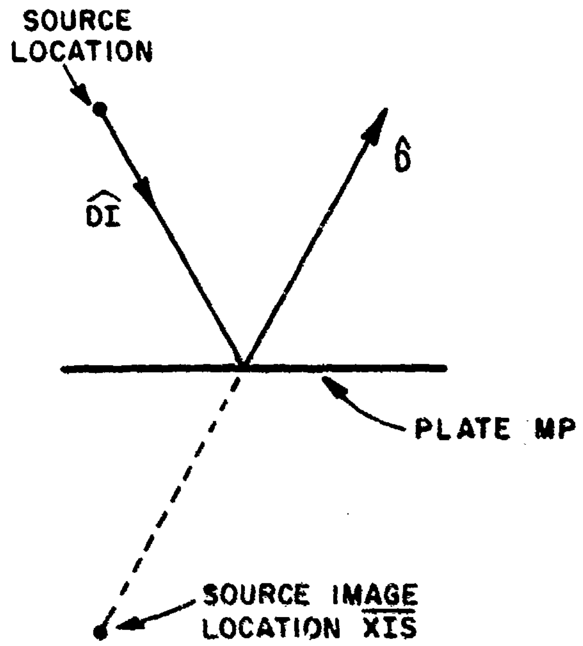


Figure 91-- Geometry for source ray reflection from plate

$$\overline{XIS} = \hat{x}\ XIS(1) + \hat{y}\ XIS(2) + \hat{z}\ XIS(3)$$

$$\hat{DI} = \hat{x}\ DI(1) + \hat{y}\ DI(2) + \hat{z}\ DI(3)$$

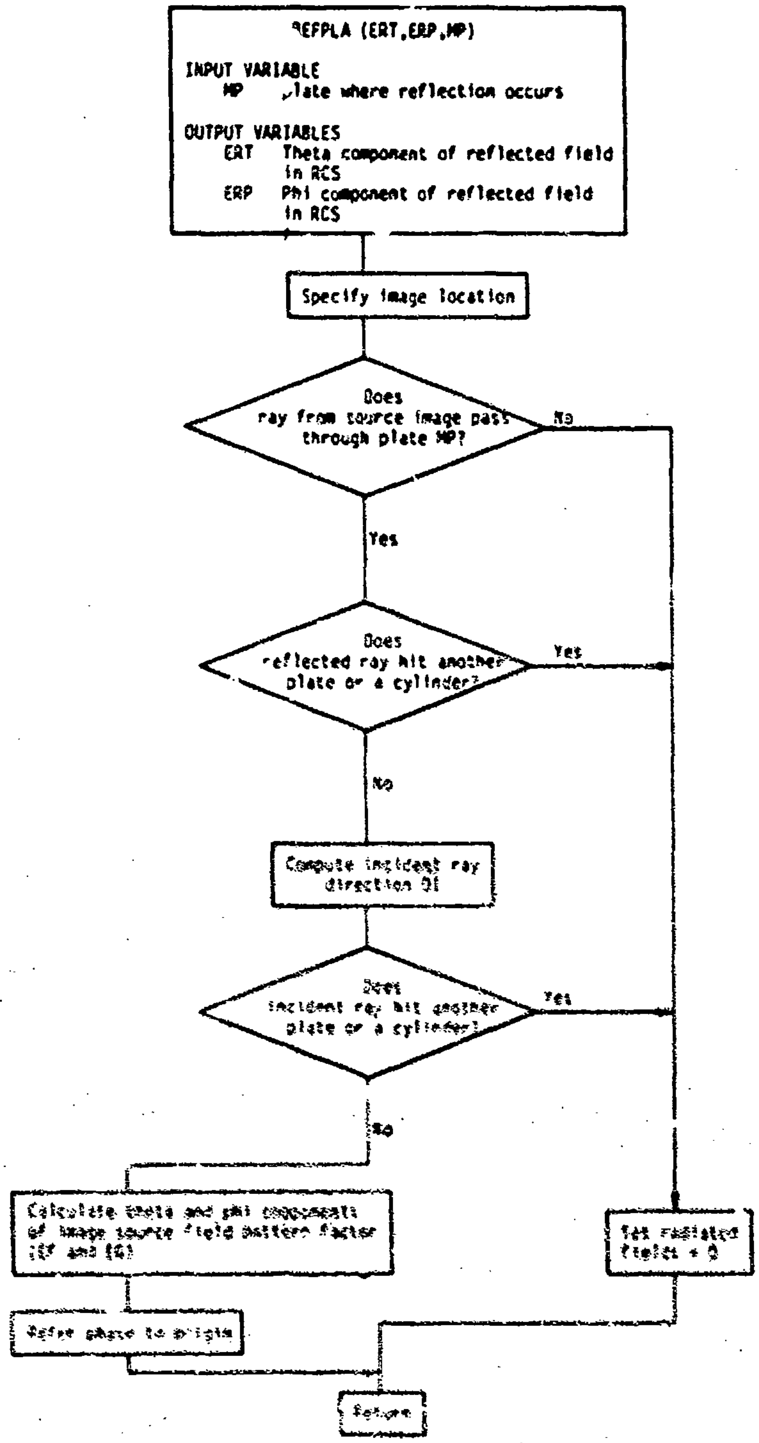
### METHOD

The reflected field from a plate is found using image theory. First the ray path is checked to insure that the reflection point is on the plate and that the ray path is not shadowed. The fields are then calculated using the SOURCE subroutine with the source coordinates oriented from image theory so that the proper boundary conditions are met at the surface of the plate. The phase is referred to the reference coordinate system origin using the factor $e^{jkO\cdot\overline{XIS}}$. The reflected field has the form

$$\bar{E}^r(r,\theta,\phi) = W_m(ERT\hat{\theta} + ERP\hat{\phi})\,\frac{e^{-jkR}}{R} \quad .$$

The factor $\frac{e^{-jkR}}{R}$ and the source current weight ($W_m$) are added elsewhere in the code.

## FLOW DIAGRAM


REFPLA (ERT,ERP,MP)
INPUT VARIABLE
MP plate where reflection occurs
OUTPUT VARIABLES
ERT Theta component of reflected field in RCS
ERP Phi component of reflected field in RCS
Specify image location
Does ray from source image pass through plate MP?
No
Yes
Does reflected ray hit another plate or a cylinder?
Yes
No
Compute incident ray direction DI
Does incident ray hit another plate or a cylinder?
Yes
No
Calculate theta and phi components of image source field pattern factor (EF and EG)
Set radiated fields = 0
Refer phase to origin
Return

## SYMBOL DICTIONARY

| | |
|---|---|
| CPHI | COSINE OF PHIR |
| CPHS | COSINE OF PHSR |
| CTHI | COSINE OF THIR |
| CTHS | COSINE OF THSR |
| D | X,Y,Z COMPONENTS OF RAY PROPAGATION DIRECTION AFTER REFLECTION IN RCS |
| DHIT | DISTANCE FROM SOURCE TO REFLECTION POINT (FROM PLAINT) |
| DHT | DISTANCE FROM SOURCE TO HIT POINT (FROM PLAINT AND CYLINT) |
| DI | X,Y,Z COMPONENTS OF INCIDENT RAY PROPAGATION DIRECTION IN RCS |
| EF | PATTERN FACTOR FOR THETA COMPONENT OF SOURCE FIELD IN RCS |
| EG | PATTERN FACTOR FOR PHI COMPONENT OF SOURCE FIELD IN RCS |
| EIX | NOT USED |
| EIY | NOT USED |
| EIZ | NOT USED |
| EX | COMPLEX PHASE FACTOR (CEXP(J*TPI*GAM)) |
| GAM | PHASE DISTANCE TO ORIGIN (DOT PRODUCT OF IMAGE LOCATION AND REFLECTED RAY PROPAGATION DIRECTION) |
| LHIT | SET TRUE IF RAY INTERSECTS A PLATE OR CYLINDER (FROM PLAINT OR CYLINT) |
| MP | PLATE FROM WHICH REFLECTION OCCURS |
| N | DO LOOP VARIABLE |
| NI | DO LOOP VARIABLE |
| NJ | DO LOOP VARIABLE |
| PHIR | PHI COMPONENT OF INCIDENT RAY PROPAGATION DIRECTION IN RCS |
| PHSR | PHI COMPONENT OF RAY PROPAGATION DIRECTION AFTER REFLECTION IN RCS |
| SPHI | SINE OF PHIR |
| SPHS | SINE OF PHSR |
| STHI | SINE OF THIR |
| THIR | THETA COMPONENT OF INCIDENT RAY PROPAGATION DIRECTION IN RCS |
| THSR | THETA COMPONENT OF RAY PROPAGATION DIRECTION AFTER REFLECTION IN RCS |
| VAX | X,Y,Z COMPONENTS DEFINING UNIT VECTORS OF THE SOURCE IMAGE COORDINATE SYSTEM AXES IN RCS |
| XI | TRIPLY DIMENSIONED ARRAY OF IMAGE LOCATIONS |
| XIS | X,Y,Z COMPONENTS OF SOURCE IMAGE LOCATION (SINGLE REFLECTION FROM PLATE MP) |
| XS | SOURCE LOCATION IN (X,Y,Z) REF COORD SYS. |

```
C-----------------------------------------------------------------------
      SUBROUTINE REFPLA(ERT,ERP,MP)
C!!!
C!!!  DETERMINES THE REFLECTED FIELD FROM PLATE #MP WITH PHASE
C!!!  REFERRED TO THE ORIGIN.
C!!!
      COMPLEX EF,EG,EX,ERT,ERP,EIX,EIY,EIZ
      DIMENSION XIS(3),DI(3),VAX(3,3)
      LOGICAL LHIT
      LOGICAL LDEBUG,LTEST
      COMMON/TEST/LDEBUG,LTEST
      COMMON/DIR/D(3),THSR,PHSR,SPHS,CPHS,STHS,CTHS
      COMMON/GEOPLA/X(14,6,3),V(14,6,3),VP(14,6,3),VN(14,3)
     2,MEP(14),MPX
      COMMON/SCRINF/XS(3),VXS(3,3)
      COMMON/IMAINF/XI(14,14,3),VXI(3,3,14)
      COMMON/PIS/PI,TPI,DPR,RPD
      IF (LDEBUG) WRITE (6,1Ø1)
 1Ø1  FORMAT (/,' DEBUGGING REFPLA SUBROUTINE')
C!!!  SPECIFY IMAGE LOCATION.
      DO 5 N=1,3
 5    XIS(N)=XI(MP,MP,N)
C!!!  DOES RAY FROM SOURCE IMAGE PASS THRU PLATE
      CALL PLAINT(XIS,D,DHIT,-MP,LHIT)
      IF(.NOT.LHIT) GO TO 3Ø
C!!!  DOES REFL. RAY HIT ANOTHER PLATE.
      CALL PLAINT(XIS,D,DHT,MP,LHIT)
      IF(LHIT) GO TO 3Ø
C!!!  DOES REFL. RAY HIT A CYLINDER.
      CALL CYLINT(XIS,D,PHSR,DHT,LHIT,.TRUE.)
      IF(LHIT) GO TO 3Ø
C!!!  KNOWING RAD. DIR. COMPUTE THE INCIDENT RAY DIRECTION
      CALL REFBP(PHIR,THIR,PHSR,THSR,MP)
      IF (LDEBUG) WRITE (6,*) PHIR,THIR,PHSR,THSR,MP
      SPHI=SIN(PHIR)
      CPHI=COS(PHIR)
      STHI=SIN(THIR)
      CTHI=COS(THIR)
      DI(1)=CPHI*STHI
      DI(2)=SPHI*STHI
      DI(3)=CTHI
C!!!  DOES RAY FROM SOURCE HIT ANOTHER PLATE.
      CALL PLAINT(XS,DI,DHT,MP,LHIT)
      IF(LHIT.AND.(DHT.LT.DHIT)) GO TO 3Ø
C!!!  DOES RAY FROM SOURCE HIT A CYLINDER.
      CALL CYLINT(XS,DI,PHIR,DHT,LHIT,.FALSE.)
      IF(LHIT.AND.(DHT.LT.DHIT)) GO TO 3Ø
      DO 2Ø NJ=1,3
      DO 2Ø NI=1,3
 2Ø   VAX(NI,NJ)=VXI(NI,NJ,MP)
C!!!  CALCULATE SOURCE FIELD PATTERN FACTOR
      CALL SOURCE(EF,EG,EIX,EIY,EIZ,THSR,PHSR,VAX)
      IF (LDEBUG) WRITE (6,*) EF,EG
C!!!  COMPUTE PHASE REFERRED TO THE ORIGIN.
      GAM=XI(MP,MP,1)*D(1)+XI(MP,MP,2)*D(2)+XI(MP,MP,3)*D(3)
      EX=CEXP(CMPLX(Ø.,TPI*GAM))
      ERT=EF*EX
      ERP=EG*EX
      GO TO 1
 3Ø   CONTINUE
      ERT=(Ø.,Ø.)
      ERP=(Ø.,Ø.)
  1   IF (.NOT.LTEST) GO TO 2
      WRITE (6,3)
  3   FORMAT (/,' TESTING REFPLA SUBROUTINE')
      WRITE (6,*) ERT,ERP,MP
  2   RETURN
      END
```

## RFDFIN

PURPOSE

To determine the reflection point on an elliptic cylinder for a given source and observation location in the near field of the cylinder.

PERTINENT GEOMETRY

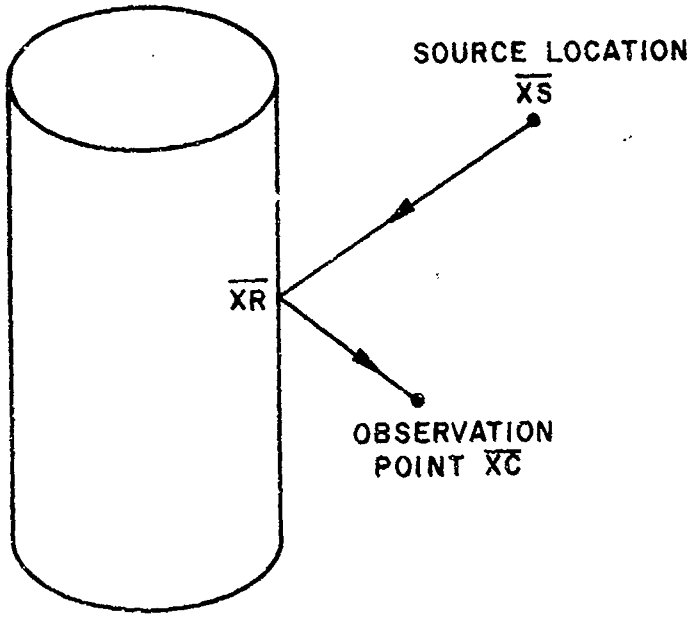


Figure 92-- Illustration of a reflection point on a cylinder for a near field observation point.

$\overline{XS} = \hat{x}\ XS(1) + \hat{y}\ XS(2) + \hat{z}\ XS(3)$

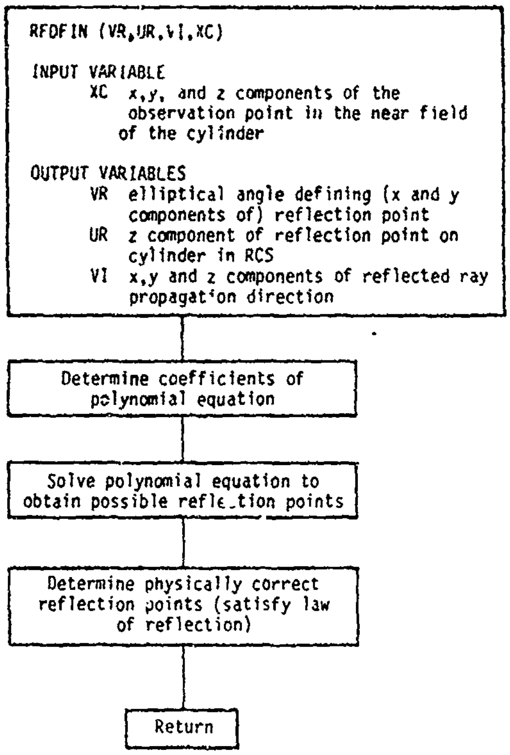
$\overline{XR} = \hat{x}\ XR(1) + \hat{y}\ XR(2) + \hat{z}\ XR(3)$

$\overline{XC} = \hat{x}\ XC(1) + \hat{y}\ XC(2) + \hat{z}\ XC(3)$

METHOD

This subroutine solves a polynomial equation, the roots which define possible reflection point locations. The true point is singled out using the laws of reflection.

## FLOW DIAGRAM


RFDFIN (VR,UR,VI,XC)
INPUT VARIABLE
XC x,y, and z components of the observation point in the near field of the cylinder
OUTPUT VARIABLES
VR elliptical angle defining (x and y components of) reflection point
UR z component of reflection point on cylinder in RCS
VI x,y and z components of reflected ray propagat'on direction
Determine coefficients of polynomial equation
Solve polynomial equation to obtain possible refle_tion points
Determine physically correct reflection points (satisfy law of reflection)
Return

## SYMBOL DICTIONARY

| | |
|---|---|
| CA | COMPLEX COEFFICIENTS OF SIXTH ORDER POLYNOMIAL EQUATION |
| RT | ROOTS OF POLYNOMIAL EQUATION |
| S | SMALLEST DISTANCE FROM SOURCE TO REFLECTION POINT TO CBSERVATION POINT |
| SM | DISTANCE FROM SOURCE TO REFLECTION POINT PLUS THE DISTANCE FROM THE REFLECTION POINT TO THE OBSERVATION POINT |
| VM | ELL ANGLE DEFINING POSSIBLE REFLECTION POINT ON CYL |
| VIN | NORMALIZATION CONSTANT FOR VI |
| XR | X,Y,Z COMPONENTS OF REFLECTION POINT LOCATION ON CYLINDER |

CODE LISTING

```
C-----------------------------------------------------------------------
      SUBROUTINE RFDFIN(VR,UR,VI,XC)
C!!!
C!!!  DETERMINES THE NEAR FIELD REFLECTION POINT FROM AN
C!!!  ELLIPTIC CYLINDER
C!!!
      COMPLEX CA(7),RT(6)
      DIMENSION XR(3),VI(3),XC(3)
      COMMON/GEOMEL/A,B,ZC(2),SNC(2),CNC(2),CTC(2)
      COMMON/SORINF/XS(3),VXS(3,3)
C!!!  DETERMINE COEFFICIENTS OF POLYNOMIAL EQUATION
      CA(7)=(A*A-B*B)*CMPLX(A*(XC(2)+XS(2)),B*(XC(1)+XS(1)))
      CA(6)=-2.*CMPLX((A*A+B*B)*(XS(1)*XC(2)+XS(2)*XC(1))
     2,2.*A*B*(A*A-B*B+XS(1)*XC(1)-XS(2)*XC(2)))
      CA(5)=CMPLX(A*(5.*B*B-A*A)*(XS(2)+XC(2))
     2,B*(5.*A*A-B*B)*(XS(1)+XC(1)))
      CA(4)=CMPLX(4.*(A*A-B*B)*(XS(1)*XC(2)+XS(2)*XC(1)),Ø.)
      CA(3)=CONJG(CA(5))
      CA(2)=CONJG(CA(6))
      CA(1)=CONJG(CA(7))
C!!!  SOLVE POLYNOMIAL EQUATION TO OBTAIN POSSIBLE
C!!!  REFLECTION POINTS
      CALL POLYRT(6,CA,RT)
      VR=BTAN2(AIMAG(RT(1)),REAL(RT(1)))
      S=SQRT((A*COS(VR)-XS(1))**2+(B*SIN(VR)-XS(2))**2)
      S=S+SQRT((XC(1)-A*COS(VR))**2+(XC(2)-B*SIN(VR))**2)
C!!!  DETERMINE PHYSICALLY CORRECT REFLECTION POINTS
C!!!  (SATISFY LAW OF REFLECTION)
      DO 1Ø I=2,6
      VM=BTAN2(AIMAG(RT(I)),REAL(RT(I)))
      XR(1)=A*COS(VM)
      XR(2)=B*SIN(VM)
      SMA=(XR(1)-XS(1))*(XR(1)-XS(1))+(XR(2)-XS(2))*(XR(2)-XS(2))
      SMB=(XC(1)-XR(1))*(XC(1)-XR(1))+(XC(2)-XR(2))*(XC(2)-XR(2))
      SM=SQRT(SMA)+SQRT(SMB)
      IF(S.LE.SM) GO TO 1Ø
      S=SM
      VR=VM
 1Ø   CONTINUE
      SNV=SIN(VR)
      CSV=COS(VR)
      XR(1)=A*CSV
      XR(2)=B*SNV
      SNX=B*CSV
      SNY=A*SNV
      SIX=XR(1)-XS(1)
      SIY=XR(2)-XS(2)
      VI(1)=XC(1)-XR(1)
      VI(2)=XC(2)-XR(2)
      SND=SNX*VI(1)+SNY*VI(2)
      SNI=SNX*SIX+SNY*SIY
      XR(3)=(SND*XS(3)-SNI*XC(3))/(SND-SNI)
      UR=XR(3)
      VI(3)=XC(3)-XR(3)
      VIM=SQRT(VI(1)*VI(1)+VI(2)*VI(2)+VI(3)*VI(3))
      DO 2Ø N=1,3
 2Ø   VI(N)=VI(N)/VIM
      RETURN
      END
```

## RFDFPT

PURPOSE

To compute the ray path for a source ray which is reflected by the cylinder and then diffracted by a given edge on a given plate.

PERTINENT GEOMETRY

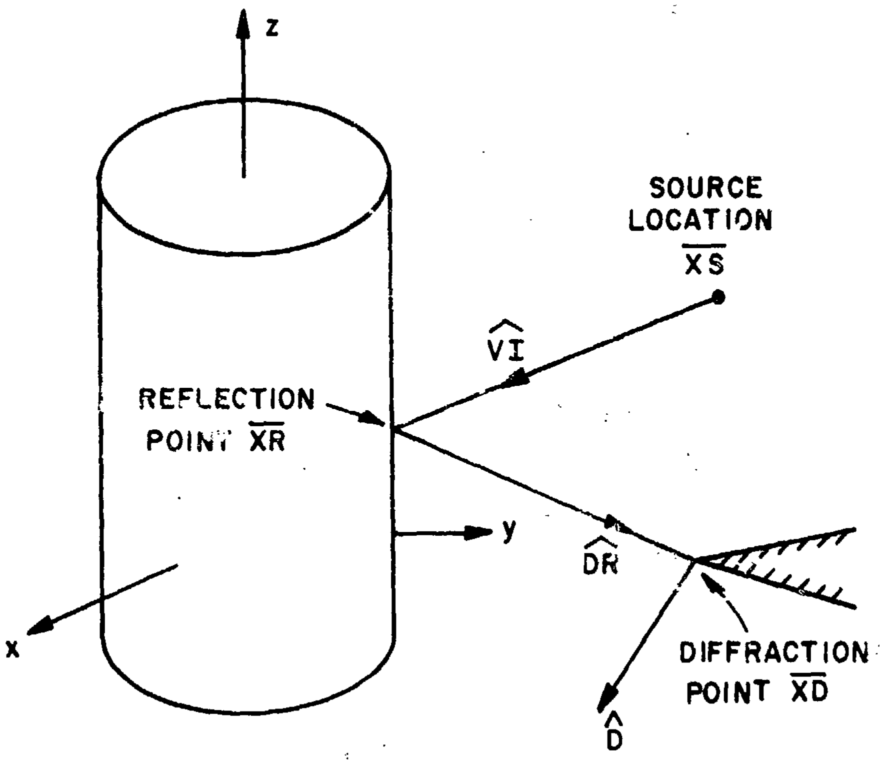

z
SOURCE LOCATION $\overline{XS}$
$\widehat{VI}$
REFLECTION POINT $\overline{XR}$
y
$\widehat{DR}$
x
DIFFRACTION POINT $\overline{XD}$
$\hat{D}$

Figure 93--Illustration of ray reflected from cylinder and then diffracted by a plate edge.

$$\overline{XR} = \hat{x}\ XR(1) + \hat{y}\ XR(2) + \hat{z}\ XR(3)$$

$$\overline{XD} = \hat{x}\ XD(1) + \hat{y}\ XD(2) + \hat{z}\ XD(3)$$

METHOD

The reflection point on an elliptic cylinder and the diffraction point on a plate edge for the reflected-diffracted ray in a given observation direction is calculated via an iterative process. The equations are based on a first order Taylor series approximation to the equations governing the laws of reflection and diffraction. The details of the analysis are given on pages 141-148 of Reference 1. The iteration process follows the same basic scheme outlined in the write up for subroutine RFPTCL. The initial start up procedure for this subroutine is composed of locating the reflection point on the cylinder for a known diffraction point which is taken to be on the corners of the plate edge under consideration. The details of this procedure are discussed on pages 149-154 of Reference 1.

## FLOW DIAGRAM

RFDFPT (VR,XR,DOTP,SNM,VIM,VI,XD,DRM,DR,
DE,ME,MP,LRDC)

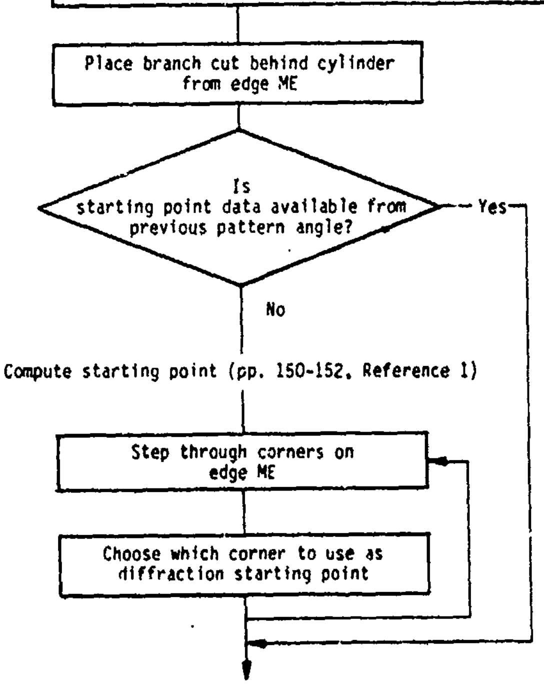
INPUT VARIABLES

MP plate where diffraction occurs
ME edge on plate ME where diffraction occurs
LRDC set true if starting point data from previous pattern cut is available
DE dot product of diffracted ray direction and edge vector of edge ME

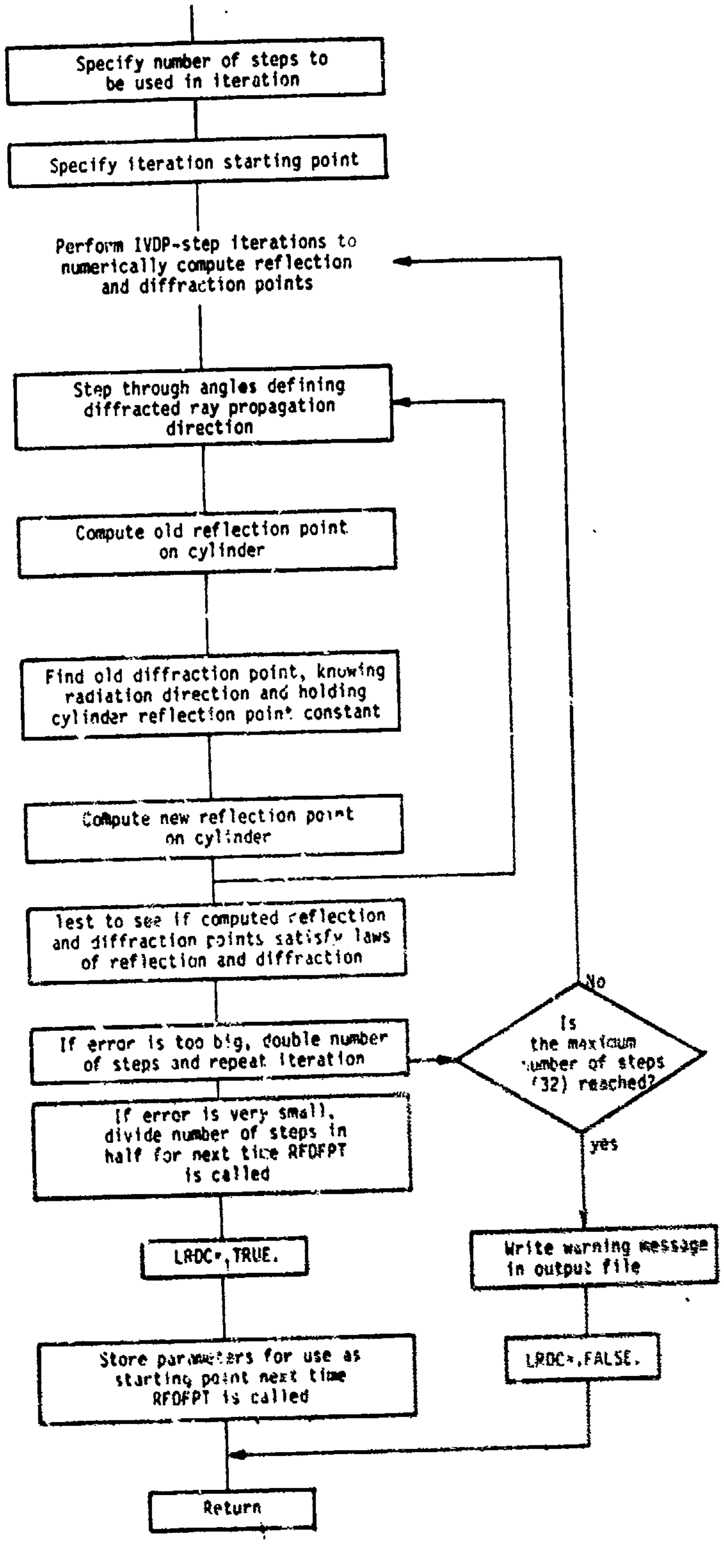
OUTPUT VARIABLES

VR elliptical angle defining reflection point on cylinder (2-D)
XR x,y,z components of reflection point on cylinder
DOTP test parameter used to determine if reflection is legal
SNM normalization constant for cylinder tangent
VIM distance from source to reflection point
VI x,y,z components of unit vector of ray from source to reflection point
XD x,y,z components of diffraction point location
DRM distance from reflection point to the diffraction point
DR unit vector of ray from reflection point to diffraction point
LRDC set true if starting point data is available for next time RFDFPT is called

Note - LRDC is used both as input and output variable



Place branch cut behind cylinder from edge ME
Is starting point data available from previous pattern angle?
Yes
No
Compute starting point (pp. 150-152, Reference 1)
Step through corners on edge ME
Choose which corner to use as diffraction starting point


Specify number of steps to be used in iteration
Specify iteration starting point
Perform IVDP-step iterations to numerically compute reflection and diffraction points
Step through angles defining diffracted ray propagation direction
Compute old reflection point on cylinder
Find old diffraction point, knowing radiation direction and holding cylinder reflection point constant
Compute new reflection point on cylinder
Test to see if computed reflection and diffraction points satisfy laws of reflection and diffraction
If error is too big, double number of steps and repeat iteration
Is the maximum number of steps (32) reached?
No
yes
If error is very small, divide number of steps in half for next time RFDFPT is called
LRDC=.TRUE.
Write warning message in output file
LRDC=.FALSE.
Store parameters for use as starting point next time RFDFPT is called
Return

# SYMBOL DICTIONARY

| | |
|---|---|
| DC | X,Y,Z COMPONENTS OF DIFFRACTED RAY PROPAGATION DIRECTION USED IN ITERATION |
| DCP | X,Y COMPONENTS OF PHI POLARIZATION UNIT VECTOR FOR DIFFRACTED RAY USED IN ITERATION |
| DCT | X,Y,Z COMPONENTS OF THETA POLARIZATION UNIT VECTOR FOR DIFFRACTED RAY USED IN ITERATION |
| DPSR | PHI ANGLE INCREMENT SIZE |
| DR | X,Y,Z COMPONENTS OF RAY DIRECTION BETWEEN REFLECTION AND DIFFRACTION |
| DRP | PARTIAL DERIVATIVE OF DR WITH RESPECT TO PHI |
| DRT | PARTIAL DERIVATIVE OF DR WITH RESPECT TO THETA |
| DRU | PARTIAL DERIVATIVE OF DR WITH RESPECT TO UR |
| DRV | PARTIAL DERIVATIVE OF DR WITH RESPECT TO VR |
| DTSR | THETA ANGLE INCREMENT SIZE |
| DU | CHANGE IN UR FOR ONE ITERATION USING TAYLOR SERIES EXPANSION |
| DV | CHANGE IN VR FOR ONE ITERATION USING TAYLOR SERIES EXPANSION |
| ERC | ERROR DETECTION VARIABLE |
| FI | EQUATION GOVERNING THE LAW OF REFLECTION |
| FP | PARTIAL DERIVATIVE OF FI WITH RESPECT TO PHI |
| FT | PARTIAL DERIVATIVE OF FI WITH RESPECT TO THETA |
| FU | PARTIAL DERIVATIVE OF FI WITH RESPECT TO UR |
| FV | PARTIAL DERIVATIVE OF FI WITH RESPECT TO VR |
| GI | EQUATION GOVERNING THE LAW OF REFLECTION |
| GP | PARTIAL DERIVATIVE OF GI WITH RESPECT TO PHI |
| GT | PARTIAL DERIVATIVE OF GI WITH RESPECT TO THETA |
| GU | PARTIAL DERIVATIVE OF GI WITH RESPECT TO UR |
| GV | PARTIAL DERIVATIVE OF GI WITH RESPECT TO VR |
| IVD | STORED NUMBER OF STEPS USED IN ITERATION |
| LRDC | SET TRUE IF STARTING POINT DATA IS AVAILABLE FROM PREVIOUS PATTERN ANGLE |
| PHCR | PHI COMPONENT OF DIFFRACTED RAY DIRECTION USED IN ITERATION |
| PHOR | PHI COMPONENT OF DIFFRACTED RAY DIRECTION FROM PREVIOUS TIME RFDFPT WAS CALLED (OR PRESENT VALUE FOR NEXT TIME ROUTINE IS CALLED) |
| PHCRP | PHI ANGLE OF DIFFRACTED RAY DIRECTION IN ROTATED RCS SYSTEM (BRANCH CUT PLACED BEHIND CYL) |
| PHSPR | PHI ANGLE OF DIFFRACTED RAY DIRECTION IN ROTATED RCS SYSTEM (BRANCH CUT PLACED BEHIND CYLINDER) |
| SNPX | PARTIAL DERIVATIVE OF SNX WITH RESPECT TO ANGLE VR |
| SNPY | PARTIAL DERIVATIVE OF SNY WITH RESPECT TO ANGLE VR |
| SNX | X COMPONENT OF NORMAL TO CYLINDER |
| SNY | Y COMPONENT OF NORMAL TO CYLINDER |
| STP | NUMBER OF STEPS USED IN ITERATION |
| THCR | THETA COMPONENT OF DIFFRACTED RAY DIRECTION USED IN ITERATION |
| THOR | THETA COMPONENT OF DIFFRACTED RAY DIRECTION FROM PREVIOUS TIME RFDFPT WAS CALLED (OR FOR NEXT TIME ROUTINE IS CALLED) |
| UR | Z COMPONENT OF REFLECTION POINT LOCATION ON CYLINDER |
| URO | STORED COMPONENTS DEFINING Z COMPONENT OF STARTING REFLECTION POINT LOCATIONS ON CYLINDER |
| VI | X,Y,Z COMPONENTS OF DIRECTION OF RAY INCIDENT ON CYLINDER |
| VIU | PARTIAL DERIVATIVE OF VI WITH RESPECT TO UR |
| VIV | PARTIAL DERIVATIVE OF VI WITH RESPECT TO ANGLE VR |
| VR | ELL ANGLE DEFINING REFLECTION POINT ON CYLINDER |
| VRO | STORED ELL ANGLES DEFINING STARTING REFLECTION POINT LOCATIONS ON CYLINDERR |
| XD | X,Y,Z COMPONENTS OF DIFFRACTION POINT LOCATION |
| XR | X,Y,Z COMPONENTS OF REFLECTION POINT LOCATION ON CYLINDER |

## CODE LISTING

```
C-----------------------------------------------------------------------
      SUBROUTINE RFDFPT(VR,XR,DOTP,SNM,VIM,VI,XD,DRM,DR,DE,ME,MP
     2,LRDC)
C!!!
C!!!  DETERMINES THE RAY PATH FOR A REFLECTION FROM THE ELLIPTIC
C!!!  CYLINDER THEN DIFFRACTION FROM A PLATE EDGE
C!!!
      DIMENSION DC(3),DCP(2),DCT(3),VI(3),VIV(3),VIU(3),VSD(3)
      DIMENSION XP(3),XR(3),XRP(3),XRV(3),XRU(3),XD(3)
      DIMENSION DR(3),DRU(3),DRV(3),DRT(3),DRP(3)
      DIMENSION IVD(14,6),PHOR(14,6),THOR(14,6),VRO(14,6),URO(14,6)
      DIMENSION PHORP(14,6)
      LOGICAL LRDC
      COMMON/GEOPLA/X(14,6,3),V(14,6,3),VP(14,6,3),VN(14,3)
     2,MEP(14),MPX
      COMMON/SORINF/XS(3),VXS(3,3)
      COMMON/DIR/D(3),THSR,PHSR,SPHS,CPHS,STHS,CTHS
      COMMON/GEOMEL/A,B,ZC(2),SNC(2),CNC(2),CTC(2)
      COMMON/LNDRCL/VCD(14,6),UCD(14,6),BCD(14,6,2)
      COMMON/FRNPHR/PHMR(14,6)
      COMMON/PIS/PI,TPI,DPR,RPD
C!!!  PLACE BRANCH CUT BEHIND CYLINDER FROM EDGE
      PHSPR=PHSR-PHMR(MP,ME)
      IF(PHSPR.GT.PI) PHSPR=PHSPR-TPI
      IF(PHSPR.LT.-PI) PHSPR=PHSPR+TPI
C!!!  IS STARTING POINT DATA AVAILABLE FROM
C!!!  PREVIOUS PATTERN ANGLE?
      IF(LRDC) GO TO 40
C!!!  COMPUTE STARTING POINT
      DOM=-2.
C!!!  STEP THRU CORNERS ON EDGE ME
C!!!  CHOOSE WHICH CORNER TO USE AS STARTING
C!!!  DIFFRACTION POINT
      DO 5 J=1,2
      MC=ME-1+J
      IF(MC.GT.MEP(MP)) MC=1
      ISGN=1
      SAM=SQRT((1.-DE*DE)/(1.-BCD(MP,ME,J)*BCD(MP,ME,J)))
    3 AM=-DE+ISGN*SAM*BCD(MP,ME,J)
      DAX=D(1)+AM*V(MP,ME,1)
      DAY=D(2)+AM*V(MP,ME,2)
      DAZ=D(3)+AM*V(MP,ME,3)
      SA=DAX*DAX+DAY*DAY
      SB=SA+DAZ*DAZ
      SA=SQRT(SA)
      SB=SQRT(SB)
      CPOP=DAX/SA
      SPOP=DAY/SA
      CTOP=DAZ/SB
      STOP=SA/SB
      DOX=CPOP*STOP
      DOY=SPOP*STOP
      DOZ=CTOP
      DOD=D(1)*DOX+D(2)*DOY+D(3)*DOZ
      DOV=DOX*V(MP,ME,1)+DOY*V(MP,ME,2)+DOZ*V(MP,ME,3)
      IF(DOV*UCD(MP,ME,J).LT.0.) GO TO 4
      IF(ABS(DOD).GT.1.) GO TO 4
      IF(DOD.LE.DOM) GO TO 4
      DOM=DOD
      JE=J
      CPO=CPOP
      SPO=SPOP
      CTO=CTOP
      STO=STOP
    4 ISGN=-ISGN
      IF(ISGN.LT.0) GO TO 3
    5      CONTINUE
           PHOR(MP,ME)=FTAN2(SPO,CPO)
           PHORP(MP,ME)=PHOR(MP,ME)-PHWR(MP,ME)
           IF(PHORP(MP,ME).GT.PI) PHORP(MP,ME)=PHORP(MP,ME)-TPI
           IF(PHORP(MP,ME).LT.-PI) PHORP(MP,ME)=PHORP(MP,ME)+TPI
           THOR(MP,ME)=FTAN2(STO,CTO)
           MC=ME-1+JE
           IF(MC.GT.MEP(MP)) MC=1
           VRO(MP,ME)=VCD(MP,MC)
           URO(MP,ME)=UCD(MP,MC)
           IVD(MP,ME)=1
C!!!       SPECIFY NUMBER OF STEPS IN ITERATION
    4      STP=IVD(MP,ME)
           IVDP=IVD(MP,ME)+1
           DPSR=(PFSPR-PHORP(MP,ME))/STP
           DTSR=(TFSR-THOR(MP,ME))/STP
C!!!       SPECIFY STARTING POINT
           VR=VRO(MP,ME)
           UR=URO(MP,ME)
C!!!       PERFORM IVDP-STEP ITERATIONS TO NUMERICALLY
C!!!       COMPUTE REFLECTION AND DIFFRACTION POINTS.
C!!!       STEP THROUGH ANGLES (DEFINING DIF. RAY PROP. DIR.)
           DO 50 IV=1,IVDP
           PHCR=PHCR(MP,ME)+(IV-1)*DPSR
           THCR=THCR(MP,ME)+(IV-1)*DTSR
           CPCS=COS(PHCR)
           SPCS=SIN(PHCR)
           CTCS=COS(THCR)
           STCS=SIN(THCR)
           DC(1)=CPCS*STCS
           DC(2)=SPCS*STCS
           DC(3)=CTCS
           DCP(1)=-SPCS*STCS
           DCP(2)=CPCS*STCS
           DCT(1)=CPCS*CTCS
           DCT(2)=SPCS*CTCS
           DCT(3)=-STCS
           CSV=COS(VR)
           SNV=SIN(VR)
           SNX=B*CSV
           SNY=A*SNV
           SNPX=-B*SNV
           SNPY=A*CSV
C!!!       COMPUTE OLD REFLECTION POINT ON CYLINDER
           XR(1)=A*CSV
           XR(2)=B*SNV
           XR(3)=UR
           XRV(1)=-A*SNV
           XRV(2)=B*CSV
           XRV(3)=0.
           XRU(1)=0.
           XRU(2)=0.
           XRU(3)=1.
           PV=0.
           DDV=0.
           DO 10 N=1,3
           VI(N)=XR(N)-XS(N)
           DDV=DDV+DC(N)*V(MP,ME,N)
   10      PV=PV+(XR(N)-X(MP,ME,N))*V(MP,ME,N)
           DO 11 N=1,3
   11      XP(N)=X(MP,ME,N)+PV*V(MP,ME,N)
           SM=0.
           DO 12 N=1,3
   12      SM=SM+(XR(N)-XP(N))*(XR(N)-XP(N))
           SM=SQRT(SM)
           COTB=DDV/SQRT(1.-DDV*DDV)
```

```
C!!!  FIND OLD DIFFRACTION POINT, KNOWING RADIATION
C!!!  DIRECTION AND HOLDING CYLINDER REFLECTION
C!!!  POINT CONSTANT
      DO 13 N=1,3
      XD(N)=XP(N)+SM*COTB*V(MP,ME,N)
      DR(N)=XD(N)-XR(N)
      VIV(N)=XRV(N)
   13 VIU(N)=XRU(N)
      IF(IV.EQ.IVDP) GO TO 60
      DDPV=DCP(1)*V(MP,ME,1)+DCP(2)*V(MP,ME,2)
      DDTV=DCT(1)*V(MP,ME,1)+DCT(2)*V(MP,ME,2)+DCT(3)*V(MP,ME,3)
      DDDV=(1.-DDV*DDV)**1.5
      CTBP=DDPV/DDDV
      CTBT=DDTV/DDDV
      DO 14 N=1,3
      DRP(N)=SM*CTBP*V(MP,ME,N)
   14 DRT(N)=SM*CTBT*V(MP,ME,N)
      CRUV=0.
      CRVV=0.
      CRUR=0.
      CRVR=0.
      CRV=0.
      DO 15 N=1,3
      CRUV=CRUV+XRU(N)*V(MP,ME,N)
      CRVV=CRVV+XRV(N)*V(MP,ME,N)
      CRUR=CRUR+XRU(N)*(XR(N)-X(MP,ME,N))
   15 CRVR=CRVR+XRV(N)*(XR(N)-X(MP,ME,N))
      CCU=CRUV+COTB*(CRUR-CRUV*PV)/SM
      CCV=CRVV+COTB*(CRVR-CRVV*PV)/SM
      DO 16 N=1,3
      DRU(N)=CCU*V(MP,ME,N)-XRU(N)
   16 DRV(N)=CCV*V(MP,ME,N)-XRV(N)
C!!!  PERFORM TAYLOR SERIES EXPANSION TO DEFINE DV AND DU
      FV=(SNPX*VI(1)+SNX*VIV(1)+SNPY*VI(2)+SNY*VIV(2))*
     2(SNX*DR(2)-SNY*DR(1))
      FV=FV+(SNPX*DR(2)+SNX*DRV(2)-SNPY*DR(1)-SNY*DRV(1))*
     2(SNX*VI(1)+SNY*VI(2))
      FV=FV+(SNPX*VI(2)+SNX*VIV(2)-SNPY*VI(1)-SNY*VIV(1))*
     2(SNX*DR(1)+SNY*DR(2))
      FV=FV+(SNPX*DR(1)+SNX*DRV(1)+SNPY*DR(2)+SNY*DRV(2))*
     2(SNX*VI(2)-SNY*VI(1))
      FU=(SNX*DR(2)-SNY*DR(1))*(SNX*VIU(1)+SNY*VIU(2))+
     2(SNX*DR(1)+SNY*DR(2))*(SNX*VIU(2)-SNY*VIU(1))
      FU=FU+(SNX*VI(1)+SNY*VI(2))*(SNX*DRU(2)-SNY*DRU(1))+
     2(SNX*DRU(1)+SNY*DRU(2))*(SNX*VI(2)-SNY*VI(1))
      GV=DR(3)*(SNPX*VI(1)+SNX*VIV(1)+SNPY*VI(2)+SNY*VIV(2))
      GV=GV+VI(3)*(SNPX*DR(1)+SNX*DRV(1)+SNPY*DR(2)+SNY*DRV(2))
      GV=GV+DRV(3)*(SNX*VI(1)+SNY*VI(2))+VIV(3)*(SNX*DR(1)+SNY*DR(2))
      GU=DR(3)*(SNX*VIU(1)+SNY*VIU(2))+VIU(3)*(SNX*DR(1)+SNY*DR(2))
      GU=GU+DRU(3)*(SNX*VI(1)+SNY*VI(2))+VI(3)*(SNX*DRU(1)+SNY*DRU(2))
      FP=(SNX*VI(1)+SNY*VI(2))*(SNX*DRP(2)-SNY*DRP(1))+
     2(SNX*VI(2)-SNY*VI(1))*(SNX*DRP(1)+SNY*DRP(2))
      FT=(SNX*VI(1)+SNY*VI(2))*(SNX*DRT(2)-SNY*DRT(1))+
     2(SNX*VI(2)-SNY*VI(1))*(SNX*DRT(1)+SNY*DRT(2))
      GP=VI(3)*(SNX*DRP(1)+SNY*DRP(2))+DRP(3)*(SNX*VI(1)+SNY*VI(2))
      GT=DRT(3)*(SNX*VI(1)+SNY*VI(2))+VI(3)*(SNX*DRT(1)+SNY*DRT(2))
      FI=(SNX*VI(1)+SNY*VI(2))*(SNX*DR(2)-SNY*DR(1))+
     2(SNX*DR(1)+SNY*DR(2))*(SNX*VI(2)-SNY*VI(1))
      GI=DR(3)*(SNX*VI(1)+SNY*VI(2))+VI(3)*(SNX*DR(1)+SNY*DR(2))
      DET=FU*GV-FV*GU
      DV=((FI*GU-GI*FU)+(GU*FP-FU*GP)*DPSR+(GU*FT-FU*GT)*DTSR)/DET
      DU=((GI*FV-FI*GV)+(FV*GP-GV*FP)*DPSR+(FV*GT-GV*FT)*DTSR)/DET
C!!!  COMPUTE NEW REFLECTION POINT ON CYLINDER
      UR=UR+DU
   40 VR=VR+DV
   50 CONTINUE
```

```
60      CONTINUE
C!!!    TEST TO SEE IF COMPUTED SCATTER POINTS SATISFY
C!!!    LAWS OF REFLECTION AND DIFFRACTION
        SNM=SQRT(SNX*SNX+SNY*SNY)
        SNX=SNX/SNM
        SNY=SNY/SNM
        DDRV=0.
        DRM=0.
        VIM=0.
        DO 20 N=1,3
        VIM=VIM+VI(N)*VI(N)
        DDRV=DDRV+DR(N)*V(MP,ME,N)
20      DRM=DRM+DR(N)*DR(N)
        VIM=SQRT(VIM)
        DRM=SQRT(DRM)
        DO 30 N=1,3
        VI(N)=VI(N)/VIM
30      DR(N)=DR(N)/DRM
        DDRV=DDRV/DRM
        ERCB=ABS(DDV-DDRV)
        SHAD=SNX*DR(1)+SNY*DR(2)
        SHADC=SNX*VI(1)+SNY*VI(2)
        ERC=SHAD+SHADC
        DOTP=.5*(SHAD-SHADC)
        ERCA=ABS(ERC)
        ERC=ERCA
        IF(ERCB.GT.ERC)ERC=ERCB
C!!!    IF ERROR IS VERY SMALL, DIVIDE NUMBER OF STEPS
C!!!    IN HALF FOR NEXT TIME ROUTINE IS CALLED
        IF(ERC.LT.0.01) GO TO 80
C!!!    IF ERROR IS TOO BIG, DOUBLE NUMBER OF STEPS
C!!!    (UP TO 32) AND REPEAT ITERATION
        IF(IVD(MP,ME).GE.32) GO TO 70
        IVD(MP,ME)=2*IVD(MP,ME)
        GO TO 40
70      CONTINUE
        WRITE(6,1) PHSR,THSR,MP,ME,VR,UR,ERCA,ERCB
1       FORMAT(' ERROR IN RFDFPT= ',2F12.6,2I5,4F12.6)
        LRDC=.FALSE.
        RETURN
80      CONTINUE
        IF(ERC.GE.0.001) GO TO 90
        IF(IVD(MP,ME).EQ.1) GO TO 90
        IVD(MP,ME)=IVD(MP,ME)/2
90      CONTINUE
C!!!    STORE PARAMETERS FOR NEXT TIME RFDFPT IS CALLED
        VRO(MP,ME)=VR
        URO(MP,ME)=UR
        PHOR(MP,ME)=PHSR
        PHORP(MP,ME)=PHSPR
        THOR(MP,ME)=THSR
        IF(.NOT.LRDC) IVD(MP,ME)=1
        LRDC=.TRUE.
        RETURN
        END
```

## RFPTCL

### PURPOSE

To calculate the reflection point on the elliptic cylinder for a source ray reflected in a given direction. The routine also computes cylinder reflection points for source rays that are reflected by a given plate and then reflected by the cylinder.

### PERTINENT GEOMETRY

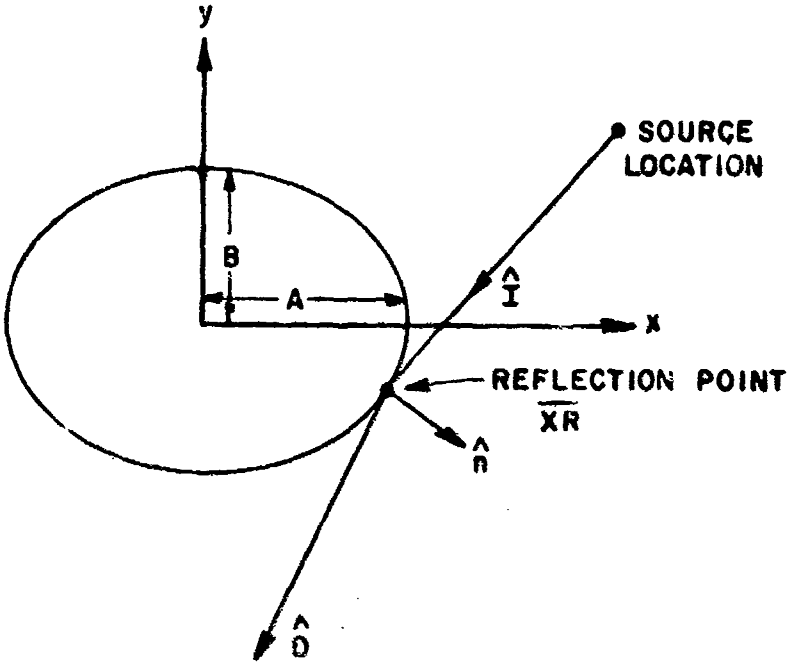


Figure 94-- Illustration of cylinder reflection point.

$\hat{I} = \hat{x}\ SIX + \hat{y}\ SIY$

$\hat{n} = \hat{x}\ SNX + \hat{y}\ SNY$

$\overline{XR} = \hat{x}\ A \cos VR + \hat{y}\ B \sin VR$

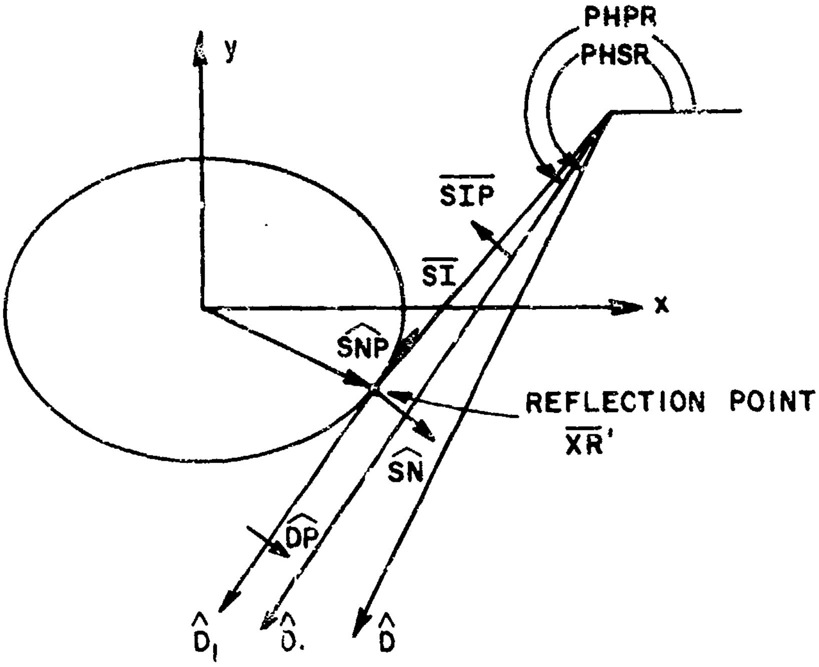


Figure 95-- Geometry for calculating reflection point.

$\hat{D}_1 = \hat{x}\ DX + \hat{y}\ DY$

$\widehat{DP} = \hat{x}\ DPX + \hat{y}\ DPY$

$\overrightarrow{SN} = \hat{x}\ SNX + \hat{y}\ SNY$

$\overrightarrow{SNP} = \hat{x}\ SNPX + \hat{y}\ SNPY$

$\overrightarrow{SI} = \hat{x}\ SIX + \hat{y}\ SIY$

$\overrightarrow{SIP} = \hat{x}\ SIPX + \hat{y}\ SIPY$

$\overline{XR}'$ = reflection point for ray with reflected phi angle PHPR
$= \hat{x}\ A\ CSV + \hat{y}\ B\ SNV$

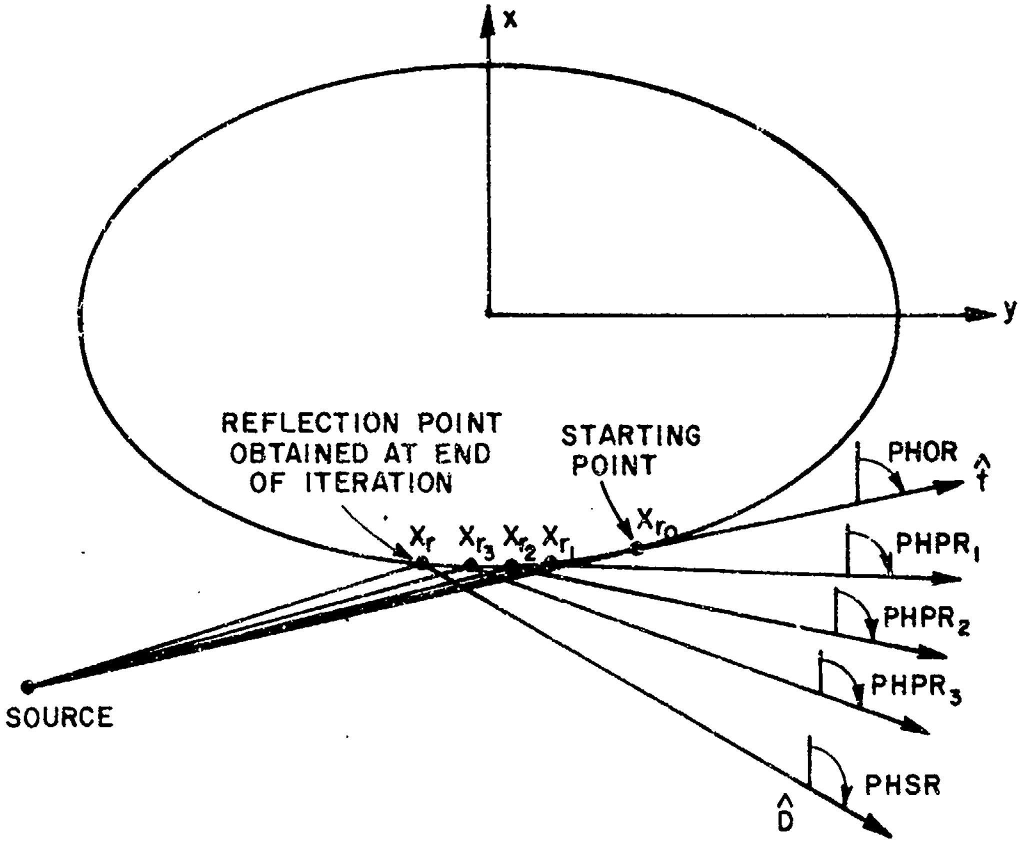


Figure 96--Illustration of iterative method used in computing the cylinder reflection point.

METHOD

The reflection point for a ray reflected in a direction defined by the phi angle PHSR is calculated via an iterative process. The routine starts with the tangent ray nearest to the reflected ray direction (or other nearby reflected ray whose reflection point is known) and steps along the cylinder surface, calculating the approximate reflection point for each reflected ray phi angle PHPR (which is stepped from PHOR to PHSR in evenly spaced steps). Each reflection point calculation uses the previous reflection point as a reference. As long as the steps are sufficiently small, the approximation is accurate. The equations are based on a first order Taylor series approximation of the equation governing the laws of reflection. Further details are given on pages 102-104 of Reference 1. The point obtained at the end of the process is the estimated reflection point. The routine then takes the sum of dot products of the cylinder normal and the incident and reflected rays (which should be zero in order to satisfy the law of reflection). If it is larger than some minimal amount, the number of steps used to iterate angle PHPR is doubled and the calculation is done again. If the error is much smaller than necessary, the number of steps used in the next calculation is divided by two.

Once a reflection point is calculated for a particular geometry. the elliptical angle defining the reflection point (VRO(MR)) is saved, along with the number of steps used to calculate it (IVD(MR)) for the next time RFPTCL is called for the same geometry. Since the next pattern angle is likely to be quite close to the previous one, this gives the computer a good starting point in defining the next reflection point, hence minimizing computer time. LRFC is a logical variable which if true tells the user that there is data from the previous pattern angle available to compute the next reflection point. If a reflection does not occur, LRFC is set false, and the next time the routine is called, it will start at the nearest tangent point.

## FLOW DIAGRAM

RFPTCL (PHSR,MP,VR,DOTP,DD,S,LRFC)

INPUT VARIABLES

PHSR phi component of reflected ray propagation direction in RCS

MP used to specify source or source image:
MP=0 indicates source
MP>0 indicates source image for reflection from plate MP

LRFC set true if reflection occurred last time subroutine RFPTCL was called

OUTPUT VARIABLES

VR elliptical angle defining 2-d reflection point in ERCS

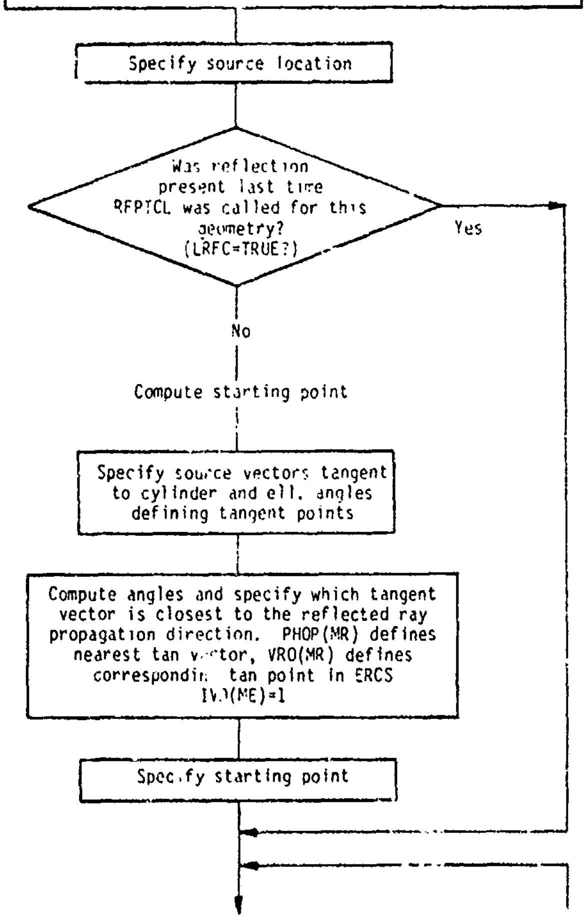
DOTP $(\hat{D}\cdot\hat{n}-\hat{I}\cdot\hat{n})/2$ (error detection variable)

DC normalization constant for $\hat{n}$, the reflection point normal

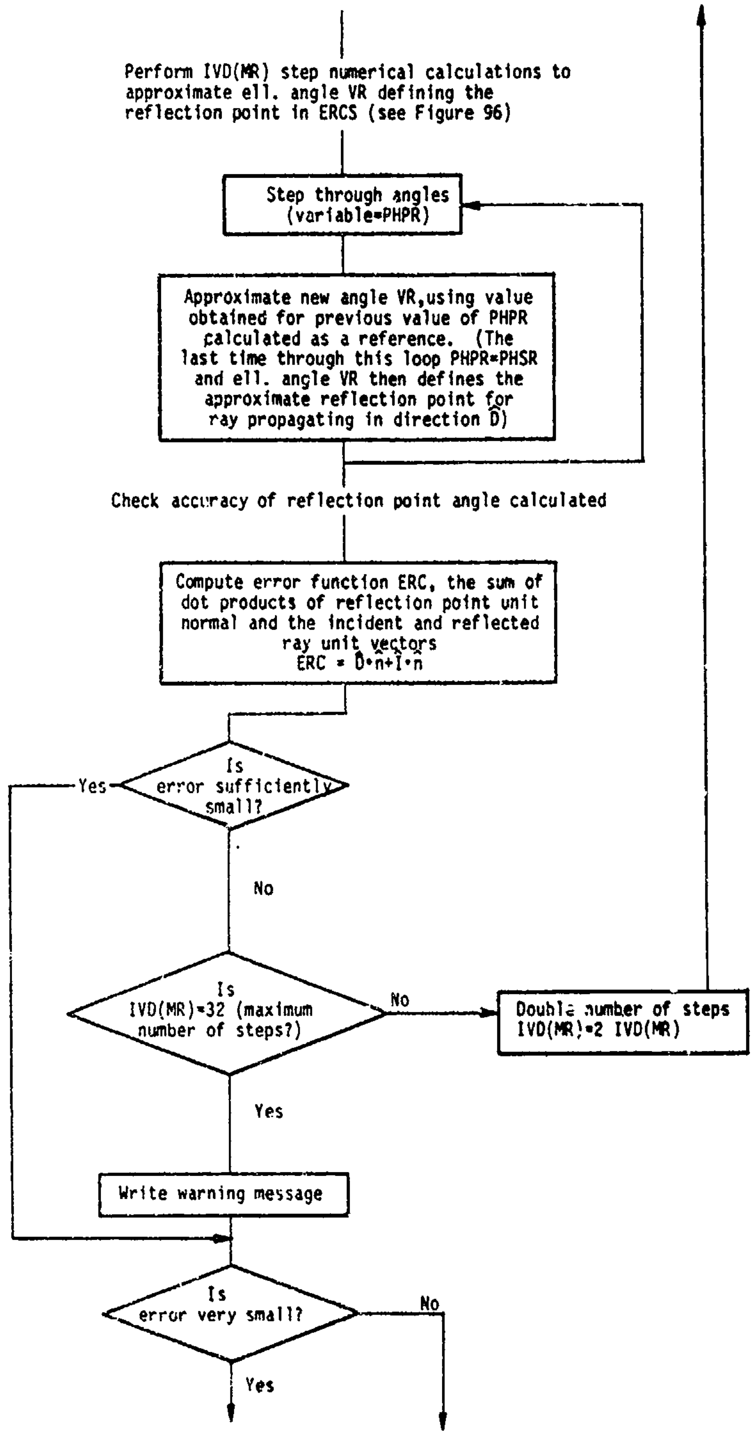
S distance from source to reflection point in x-y plane

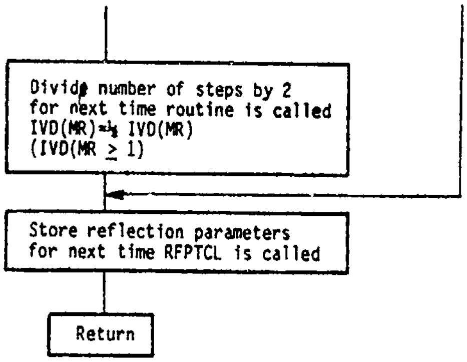
LRFC set true to indicate presence of stored starting point data for next time RFPTCL is called

Note: LRFC is used both to input and output data



Specify source location
Was reflection present last time RFPTCL was called for this geometry? (LRFC=TRUE?)
Yes
No
Compute starting point
Specify source vectors tangent to cylinder and ell. angles defining tangent points
Compute angles and specify which tangent vector is closest to the reflected ray propagation direction. PHOP(MR) defines nearest tan vector, VRO(MR) defines corresponding tan point in ERCS IVO(ME)=1
Specify starting point


Perform IVD(MR) step numerical calculations to approximate ell. angle VR defining the reflection point in ERCS (see Figure 96)
Step through angles (variable=PHPR)
Approximate new angle VR, using value obtained for previous value of PHPR calculated as a reference. (The last time through this loop PHPR=PHSR and ell. angle VR then defines the approximate reflection point for ray propagating in direction D̂)
Check accuracy of reflection point angle calculated
Compute error function ERC, the sum of dot products of reflection point unit normal and the incident and reflected ray unit vectors ERC = D̂•n̂+Î•n̂
Is error sufficiently small?
Yes
No
Is IVD(MR)=32 (maximum number of steps?)
No
Double number of steps IVD(MR)=2 IVD(MR)
Yes
Write warning message
Is error very small?
No
Yes


Divide number of steps by 2
for next time routine is called
IVD(MR)=½ IVD(MR)
(IVD(MR ≥ 1)
Store reflection parameters
for next time RFPTCL is called
Return

## SYMBOL DICTIONARY

| | |
|---|---|
| CPP | COSINE OF PHPR |
| CPS | COSINE OF PHSR |
| CSV | COSINE OF VR |
| DD | NORMALIZATION CONSTANT FOR REFL POINT NORMAL VECTOR |
| DOTP | ONE HALF THE DIFFERENCE BETWEEN THE DOT PRODUCTS OF THE REFLECTED RAY DIRECTION AND CYLINDER UNIT NORMAL AND THE INCIDENT RAY DIRECTION AND CYLINDER UNIT NORMAL |
| DPX<br>DPY | X AND Y COMPONENTS OF PARTIAL DERIVATIVE OF REFLECTED RAY DIRECTION WITH RESPECT TO PHI OBSERVATION ANGLE |
| DR | DOT PRODUCT OF INC RAY UNIT VECTOR AND CYL UNIT NORMAL |
| DS | DOT PRODUCT OF REFLECTED RAY PROPAGATION DIRECTION UNIT VECTOR AND CYLINDER UNIT NORMAL |
| DSPR | SIZE OF ANGLE STEP USED IN ITERATION |
| DV | CHANGE IN ANGLE VR |
| DX<br>DY | X AND Y COMPONENTS OF UNIT VECTOR OF REFLECTED RAY (DIRECTION DEFINED BY ANGLE PHPR) IN RCS |
| DVB | PARTIAL DERIVATIVE OF THE REFLECTION LAW EQUATION (FI) WITH RESPECT TO ELL ANGLE V |
| DVT | PARTIAL DERIVATIVE OF THE REFLECTION LAW EQUATION (FI) WITH RESPECT TO THE PHI ANGLE OF THE OBSERVATION DIRECTION |
| ERC | ERROR PARAMETER (SUM OF DS AND DR) |
| ERCA | ABSOLUTE VALUE OF ERC |
| ERCS | (NOT A VARIABLE) ABBREVIATION FOR ELLIPTICAL REFERENCE COORDINATE SYSTEM |
| FI | EQUATION SATISFYING THE LAW OF REFLECTION |
| IVD | NUMBER OF ITERATIONS USED TO FIND REFL POINT THE LAST TIME RFPTCL WAS CALLED FOR PLATE MP |
| IVDM | NUMBER OF STEPS USED IN ITERATION |
| LRFC | (ENTERING ROUTINE) SET TRUE IF REFL OCCURED LAST TIME REFCYL WAS CALLED. (LRFC ALWAYS SET TRUE LEAVING ROUTINE) |
| MP | USED TO SPECIFY WHETHER SOURCE OR SOURCE IMAGE IS USED<br>MP=Ø DESIGNATES SOURCE<br>MP>Ø DESIGNATES SOURCE IMAGE FOR REFLECTION FROM PLATE MP |
| MR | INDEX VARIABLE (MP+MPRX+1) FOR STORING DATA FOR NEXT CALL TO RFPTCL |
| PHE | PHI ANGLE BETWEEN REFLECTED RAY DIRECTION AND TANGENT POINT #2 |
| PHEP | PHI ANGLE BETWEEN REFLECTED RAY DIRECTION AND TANGENT POINT #1 |
| PHIR | PHI COMPONENT OF SOURCE LOCATION IN RCS |
| PHOR | REFLECTED RAY PHI ANGLE (STORED AS STARTING POINT PARAMETER FOR NEXT TIME ROUTINE IS CALLED) |
| PHORB | PHI ANGLE DEFINING RAY TANGENT TO TAN POINT 1 |
| PHORP | PHI ANGLE OF CYLINDER REFLECTED RAY DIRECTION IN ROTATED RCS SYSTEM |
| PHPR | REFLECTED RAY PHI ANGLE (ITERATED FROM PHOR TO PHSR) |
| PHSPR | PHI ANGLE DEFINING REFLECTED RAY DIRECTION IN ROTATED RCS |
| PHSR | PHI COMPONENT OF REFLECTED RAY PROPAGATION DIRECTION IN RCS |
| S | DISTANCE FROM SOURCE TO REFL POINT IN X-Y PLANE |
| SIPX<br>SIPY | X AND Y COMPONENTS OF PARTIAL DERIVATIVE OF INCIDENT RAY VECTOR WITH RESPECT TO ELL ANGLE V |
| SIX<br>SIY | X AND Y COMPONENTS OF INCIDENT RAY PROP VECTOR IN RCS (NOT CONSISTANTLY NORMALIZED) |
| SNPX<br>SNPY | X AND Y COMPONENTS OF PARTIAL DERIVATIVE OF CYLINDER NORMAL AT REFLECTION POINT WITH RESPECT TO ELL ANGLE V |
| SNV | SINE OF VR |
| SNX<br>SNY | X AND Y COMPONENTS OF RAY NORMAL TO CYL REFL POINT IN RCS (NOT CONSISTANTLY NORMALIZED) |
| SPP | SINE OF PHPR |
| SPS | SINE OF PHSR |
| STP | NUMBER OF STEPS USED IN ITERATION |

| | |
|---|---|
| VR | ELL. ANGLE DEFINING REFL POINT IN ERCS |
| VRO | ELL ANGLES DEFINING TANGENT POINTS FOR SOURCE RAY (OR SOURCE RAY REFLECTED FROM PLATE) TANGENT TO CYLINDER |
| XIS | SOURCE LOCATION |

# CODE LISTING

```
C-------------------------------------------------------------------------
      SUBROUTINE RFPTCL(PHSR,MP,VR,DOTP,DD,S,LRFC)
C!!!
C!!!  DETERMINES REFLECTION POINT ON AN ELLIPTIC CYLINDER
C!!!
      LOGICAL LRFC,LGRND
      DIMENSION IVD(29),PHOR(29),VRO(29),XIS(3),PHORP(29)
      COMMON/GEOMEL/A,B,ZC(2),SNC(2),CNC(2),CTC(2)
      COMMON/SORINF/XS(3),VXS(3,3)
      COMMON/GEOPLA/X(14,6,3),V(14,6,3),VP(14,6,3),VN(14,3)
     2,MEP(14),MPX
      COMMON/IMAINF/XI(14,14,3),VXI(3,3,14)
      COMMON/PIS/PI,TPI,DPR,RPD
      COMMON/ENDSCL/DTS,VTS(2),BTS(4)
      COMMON/ENDICL/DTI(14),VTI(14,2),BTI(14,4)
      COMMON/GROUND/LGRND,MPXR
      MR=MP+MPXR+1
      SPS=SIN(PHSR)
      CPS=COS(PHSR)
C!!!  SPECIFY SOURCE LOCATION
      IF(MP.GT.Ø) GO TO 11
      DO 1Ø N=1,3
 1Ø   XIS(N)=XS(N)
      PHIR=BTAN2(XS(2),XS(1))
      GO TO 15
 11   CONTINUE
      DO 12 N=1,3
 12   XIS(N)=XI(MP,MP,N)
      PHIR=BTAN2(XI(MP,MP,2),XI(MP,MP,1))
 15   CONTINUE
      PHSPR=PHSR-PHIR
      IF(PHSPR.GT.PI) PHSPR=PHSPR-TPI
      IF(PHSPR.LT.-PI) PHSPR=PHSPR+TPI
C!!!  WAS REFLECTION PRESENT LAST TIME REFCYL WAS CALLED?
      IF(LRFC) GO TO 4Ø
      IVD(MR)=1
C!!!  SPECIFY TANGENT VECTORS
      IF(MP.GT.Ø) GO TO 2Ø
      PHOR(MR)=BTAN2(BTS(4),BTS(3))
      VRO(MR)=VTS(2)
      PHORB=BTAN2(BTS(2),BTS(1))
      GO TO 25
 2Ø   CONTINUE
      PHOR(MR)=BTAN2(BTI(MP,4),BTI(MP,3))
      VRO(MR)=VTI(MP,2)
      PHORB=BTAN2(BTI(MP,2),BTI(MP,1))
 25   CONTINUE
C!!!  COMPUTE ANGLES AND SPECIFY WHICH TAN VECTOR IS CLOSER
C!!!  TO THE REFL PROPAGATION DIRECTION
      PHORP(MR)=PHOR(MR)-PHIR
      IF(PHORP(MR).GT.PI) PHORP(MR)=PHORP(MR)-TPI
      IF(PHORP(MR).LT.-PI) PHORP(MR)=PHORP(MR)+TPI
      PHORBP=PHORB-PHIR
      IF(PHORBP.GT.PI) PHORBP=PHORBP-TPI
      IF(PHORBP.LT.-PI) PHORBP=PHORBP+TPI
      PHE=ABS(PHSPR-PHORP(MR))
      PHEP=ABS(PHSPR-PHORBP)
      IF(PHEP.GE.PHE) GO TO 4Ø
      PHOR(MR)=PHORB
      PHORP(MR)=PHORBP
      VRO(MR)=VTS(1)
      IF(MP.GT.Ø) VRO(MR)=VTI(MP,1)
C!!!  INCREMENT ANGLE PHPR FROM THE CYL TAN ANGLE PHOR TO
C!!!  PROP. ANGLE PHSR IN IVD(MR) STEPS AND CALCULATE APPROX.
C!!!  VR (THE ELL. ANGLE DEFINING THE REFL POINT) FOR EACH
C!!!  ANGLE PHPR UNTIL PHPR=PHSR AND APPROX VR FOR REFL POINT
```

```
C!!!  IN DESIRED PROP. DIRECTION IS OBTAINED.
40    STP=IVD(MR)
      DPSR=(PHSPR-PHORP(MR))/STP
      VR=VRO(MR)
      IVDM=IVD(MR)
C!!!  STEP THRU ANGLES
      DO 50 IV=1,IVDM
      PHPR=PHOR(MR)+(IV-1)*DPSR
      CPP=COS(PHPR)
      SPP=SIN(PHPR)
      DX=CPP
      DY=SPP
      DPX=-SPP
      DPY=CPP
      CSV=COS(VR)
      SNV=SIN(VR)
      SNX=B*CSV
      SNY=A*SNV
      SIX=A*CSV-XIS(1)
      SIY=B*SNV-XIS(2)
      SNPX=-B*SNV
      SNPY=A*CSV
      SIPX=-A*SNV
      SIPY=B*CSV
      FI=(SNX*SIX+SNY*SIY)*(SNX*DY-SNY*DX)+
     2(SNX*DX+SNY*DY)*(SNX*SIY-SNY*SIX)
      DVT=(SNX*SIX+SNY*SIY)*(SNX*DPY-SNY*DPX)
      DVT=DVT+(SNX*DPX+SNY*DPY)*(SNX*SIY-SNY*SIX)
      DVB=(SNPX*SIX+SNX*SIPX+SNPY*SIY+SNY*SIPY)*
     2(SNX*DY-SNY*DX)
      DVB=DVB+(SNPX*SIY+SNX*SIPY-SNPY*SIX-SNY*SIPX)*
     2(SNX*DX+SNY*DY)*
      DVB=DVB+(SNX*SIX+SNY*SIY)*(SNPX*DY-SNPY*DX)
      DVB=DVB+(SNPX*DX+SNPY*DY)*(SNX*SIY-SNY*SIX)
      DV=-(FI+DVT*DPSR)/DVB
C!!!  APPROXIMATE ANGLE VR FOR THE REFL POINT IN DIRECTION PHPR
      VR=VR+DV
50    CONTINUE
C!!!  CHECK ACCURACY OF REFLECTION POINT ANGLE CALCULATED
      CSV=COS(VR)
      SNV=SIN(VR)
      SNX=B*CSV
      SNY=A*SNV
      DD=SQRT(SNX*SNX+SNY*SNY)
      SNX=SNX/DD
      SNY=SNY/DD
      SIX=A*CSV-XIS(1)
      SIY=B*SNV-XIS(2)
      S=SQRT(SIX*SIX+SIY*SIY)
      SIX=SIX/S
      SIY=SIY/S
C!!!  CALCULATE THE ERROR FUNCTION ERC, THE SUM OF DOT
C!!!  PRODUCTS OF INCIDENT AND REFLECTED UNIT VECTORS AND
C!!!  CYLINDER UNIT NORMAL (SHOULD BE CLOSE TO ZERO)
      DS=SNX*CPS+SNY*SPS
      DR=SNX*SIX+SNY*SIY
      DOTP=.5*(DS-DR)
      ERC=DS+DR
      ERCA=ABS(ERC)
C!!!  IF ERROR IS NOT SUFFICIENTLY SMALL, DOUBLE NUMBER OF STEPS
C!!!  (UP TO 32) AND RECALCULATE VR
      IF(ERCA.LT.0.0005) GO TO 80
C!!!  IF MAX NUMBER OF STEPS ALREADY REACHED, PRINT WARNING
      IF(IVD(MR).GE.32) GO TO 70
      IVD(MR)=2*IVD(MR)
      GO TO 40
```

```
70      CONTINUE
        WRITE(6,1) ERC,VR,PHSR
1       FORMAT(' ERROR IN RFPTCL= ',3F12.6)
80      CONTINUE
C!!!    IF ERROR IS VERY SMALL, DIVIDE NUMBER OF ITERATION
C!!!    STEPS USED IN HALF FOR NEXT TIME ROUTINE IS CALLED
        IF(ERCA.GE.0.00005) GO TO 90
        IF(IVD(MR).EQ.1) GO TO 90
        IVD(MR)=IVD(MR)/2
90      CONTINUE
        VRO(MR)=VR
        PHOR(MR)=PHSR
        PHORP(MR)=PHSPR
        IF(.NOT.LRFC) IVD(MR)=1
        LRFC=.TRUE.
        RETURN
        END
```

## ROTRAN

### PURPOSE

To transform a point or vector defined in the old reference coordinate system to the new (cylinder-centered) reference coordinate system representation. This is used in the main program to perform the reference coordinate system transformation.

### PERTINENT GEOMETRY

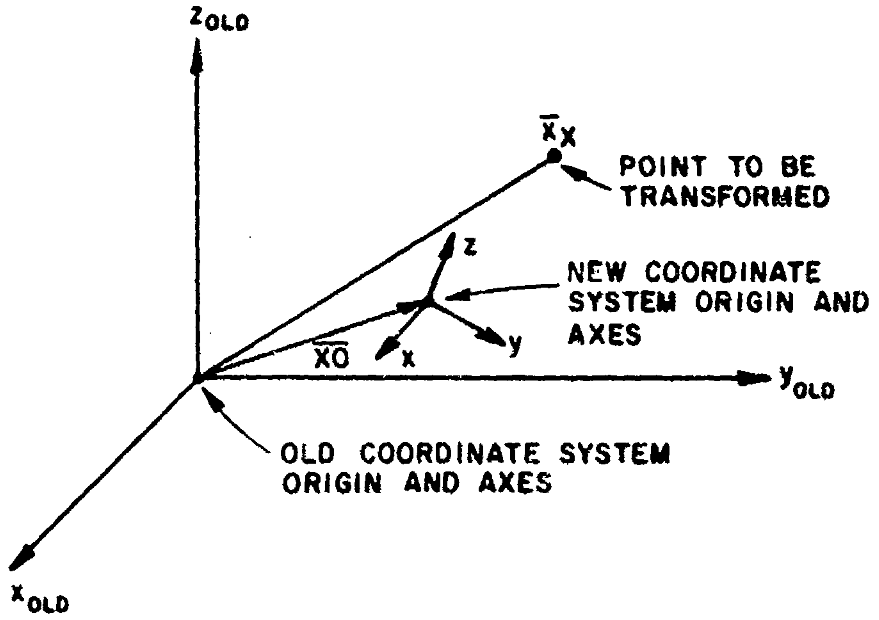


Figure 97-- Illustration of old and new reference coordinate systems.

$$\overline{X}_x = \hat{x}_{old}\ XX(1) + \hat{y}_{old}\ XX(2) + \hat{z}_{old}\ XX(3)$$

$$\overline{X}_x = \hat{x}\ XRT(1) + \hat{y}\ XRT(2) + \hat{z}\ XRT(3)$$

### METHOD

The point $\overline{X}_x$ defined in the old coordinate system may be represented by point $\overline{X}_{rt}$ in the new cooridnate system where:

$$\overline{X}_{rt} = \left[V_{cl}\right]\overline{X}_t, \text{ where } \overline{X}_t = \overline{X}_x - \overline{X}_o$$

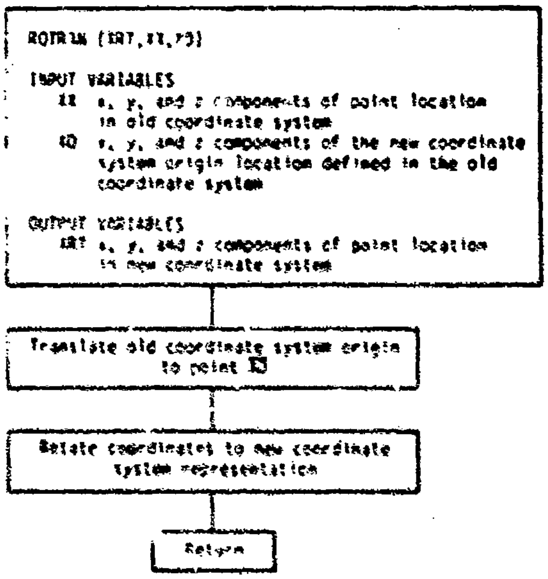
or

$$\begin{bmatrix} XRT(1) \\ XRT(2) \\ XRT(3) \end{bmatrix} = \begin{bmatrix} XCL(1) & XCL(2) & XCL(3) \\ YCL(1) & YCL(2) & YCL(3) \\ ZCL(1) & ZCL(2) & ZCL(3) \end{bmatrix} \begin{bmatrix} XX(1) - XO(1) \\ XX(2) - XO(2) \\ XX(3) - XO(3) \end{bmatrix} ,$$

where $\overline{X}_o$ is the location of the new coordinate system origin defined in the old coordinate system and $\hat{x}$, $\hat{y}$, $\hat{z}$ are unit vectors defining the new coordinate system axes in old coordinate system components:

$$\hat{x} = \hat{x}_{o'd}\, XCL(1) + \hat{y}_{old}\, XCL(2) + \hat{z}_{old}\, XCL(3)$$
$$\hat{y} = \hat{x}_{old}\, YCL(1) + \hat{y}_{old}\, YCL(2) + \hat{z}_{old}\, YCL(3)$$
$$\hat{z} = \hat{x}_{old}\, ZCL(1) + \hat{y}_{old}\, ZCL(2) + \hat{z}_{old}\, ZCL(3).$$

FLOW DIAGRAM


ROTRAN (XRT,XX,XO)
INPUT VARIABLES
XX x, y, and z components of point location in old coordinate system
XO x, y, and z components of the new coordinate system origin location defined in the old coordinate system
OUTPUT VARIABLES
XRT x, y, and z components of point location in new coordinate system
Translate old coordinate system origin to point XO
Rotate coordinates to new coordinate system representation
Return

## SYMBOL DICTIONARY

XT     X,Y, AND Z COMPONENTS OF POINT LOCATION AFTER TRANSLATING OLD COORDINATE SYSTEM ORIGIN TO POINT X0

## CODE LISTING

```
C-----------------------------------------------------------------------
      SUBROUTINE ROTRAN(XRT,XX,XO)
C!!!
C!!!  COORDINATE TRANSLATION AND ROTATION: XO IS THE
C!!!  NEW ORIGIN: XCL,YCL,ZCL DEFINE THE NEW AXES.
C!!!
      DIMENSION XRT(3),XX(3),XO(3),XT(3)
      LOGICAL LDEBUG,LTEST
      COMMON/ROTROT/XCL(3),YCL(3),ZCL(3)
      COMMON/TEST/LDEBUG,LTEST
C!!!  TRANSLATION OF COORDINATES
      DO 10 N=1,3
   10 XT(N)=XX(N)-XO(N)
C!!!  ROTATION OF COORDINATES
      XRT(1)=XT(1)*XCL(1)+XT(2)*XCL(2)+XT(3)*XCL(3)
      XRT(2)=XT(1)*YCL(1)+XT(2)*YCL(2)+XT(3)*YCL(3)
      XRT(3)=XT(1)*ZCL(1)+XT(2)*ZCL(2)+XT(3)*ZCL(3)
      IF(.NOT.LTEST) RETURN
      WRITE(6,900)
  900 FORMAT(/,' TESTING ROTRAN SUBROUTINE')
      WRITE(6,*) XRT
      WRITE(6,*) XX
      WRITE(6,*) XO
      RETURN
      END
```

RPLDPL

PURPOSE

To calculate the far-zone electric field (with phase referred to the RCS origin) for a source ray that reflects off plate MR and is then diffracted off edge ME of plate MP.

PERTINENT GEOMETRY

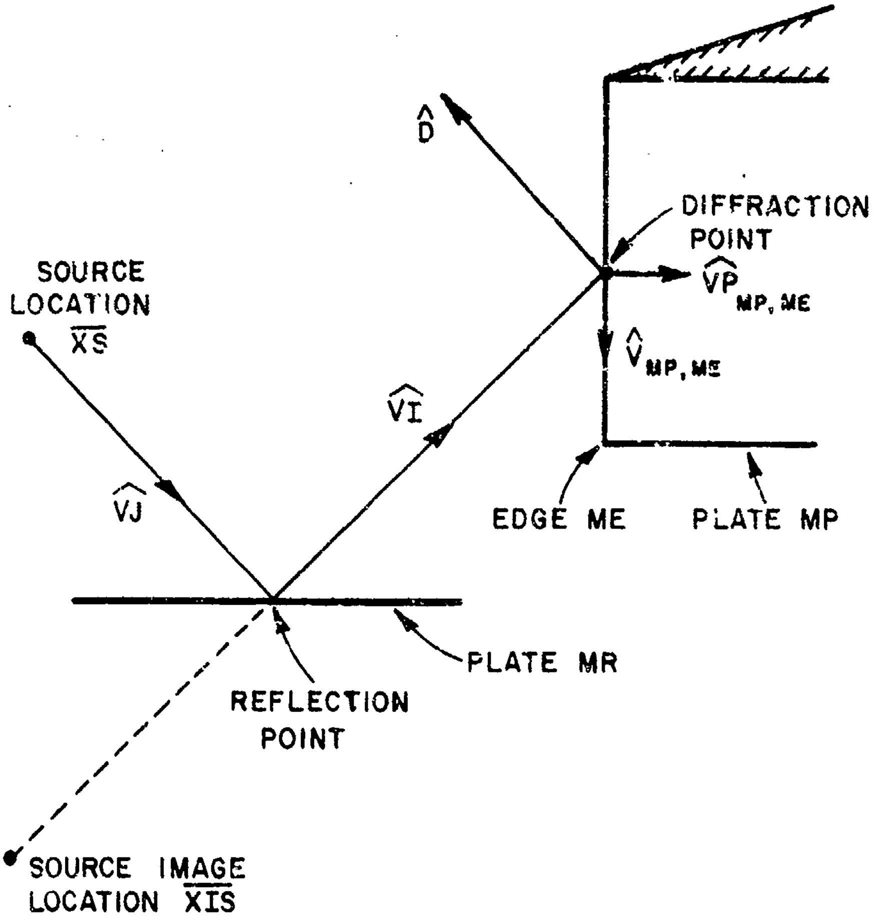


Figure 98--Illustration of a ray reflected by a plate and then diffracted by a plate edge.

## METHOD

The field reflected by a plate and then diffracted by another plate edge is calculated in this subroutine [4,9,10]. The field reflected from the plate is found using image thoery. The diffracted and slope diffracted fields of the plate edges and corners are obtained as described in subroutine DIFPLT. The diffracted edge and slope fields are combined and the phase is referred to the reference coordinate system origin by the factor $e^{jk\hat{D}\cdot\overline{XDP}}$. The form of the field is therefore given by

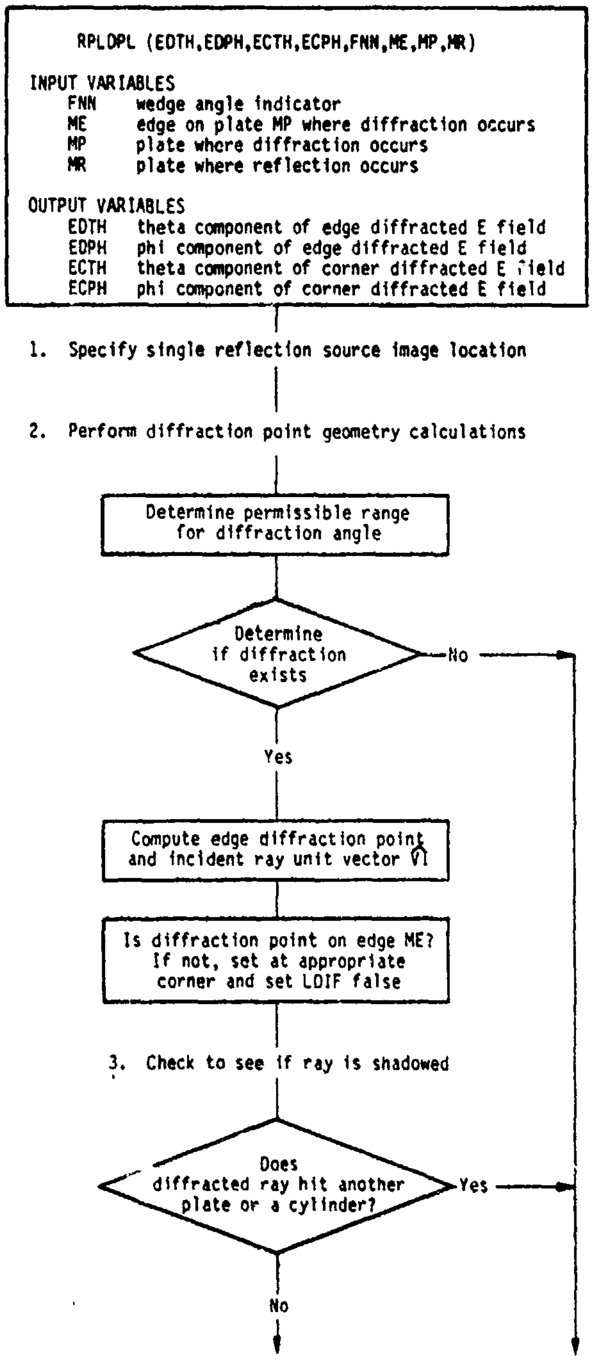
$$\overline{E}^d = W_m(EDTH\hat{\theta} + EDPH\hat{\phi})\,\frac{e^{-jkR}}{R} \quad .$$

The corner and slope corner diffracted fields are combined in a similar way and are given by

$$\overline{E}^c = W_m(ECTH\hat{\theta} + ECPH\hat{\phi})\,\frac{e^{-jkR}}{R}$$

where the factor $\frac{e^{-jkR}}{R}$ and the source ($W_m$) weight are added elsewhere in the code.

## FLOW DIAGRAM



RPLDPL (EDTH,EDPH,ECTH,ECPH,FNN,ME,MP,MR)

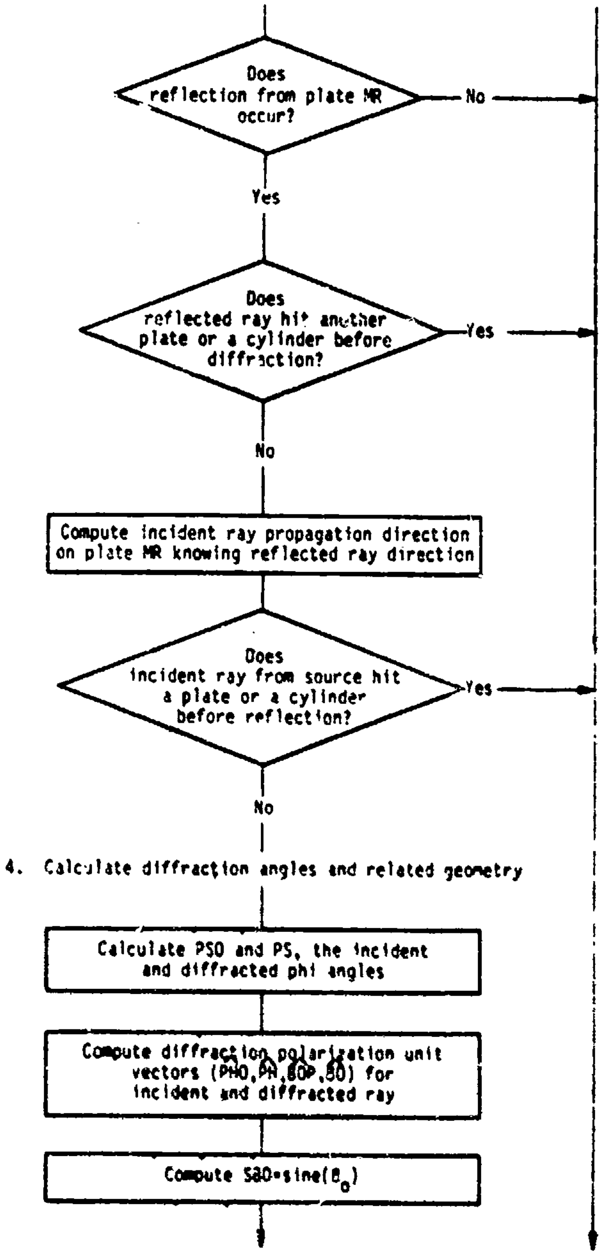
INPUT VARIABLES

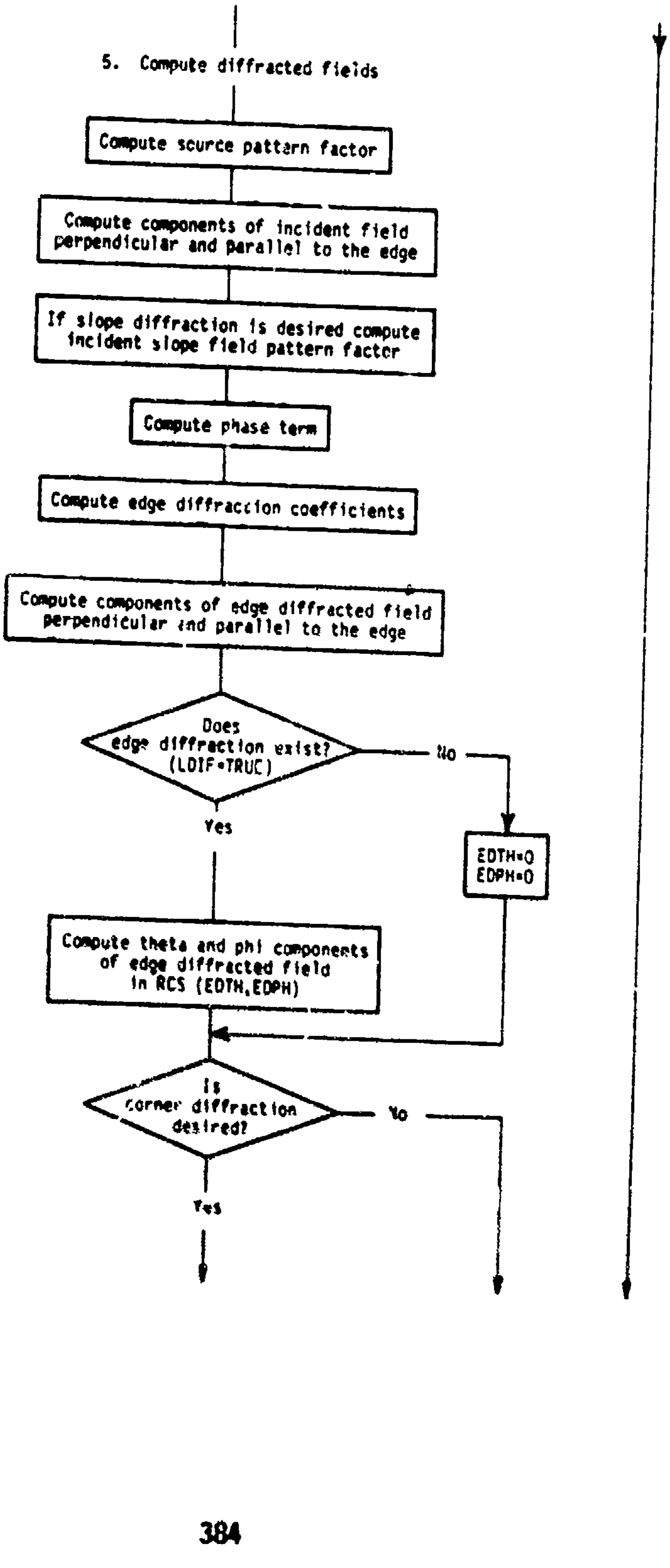
| | |
|---|---|
| FNN | wedge angle indicator |
| ME | edge on plate MP where diffraction occurs |
| MP | plate where diffraction occurs |
| MR | plate where reflection occurs |

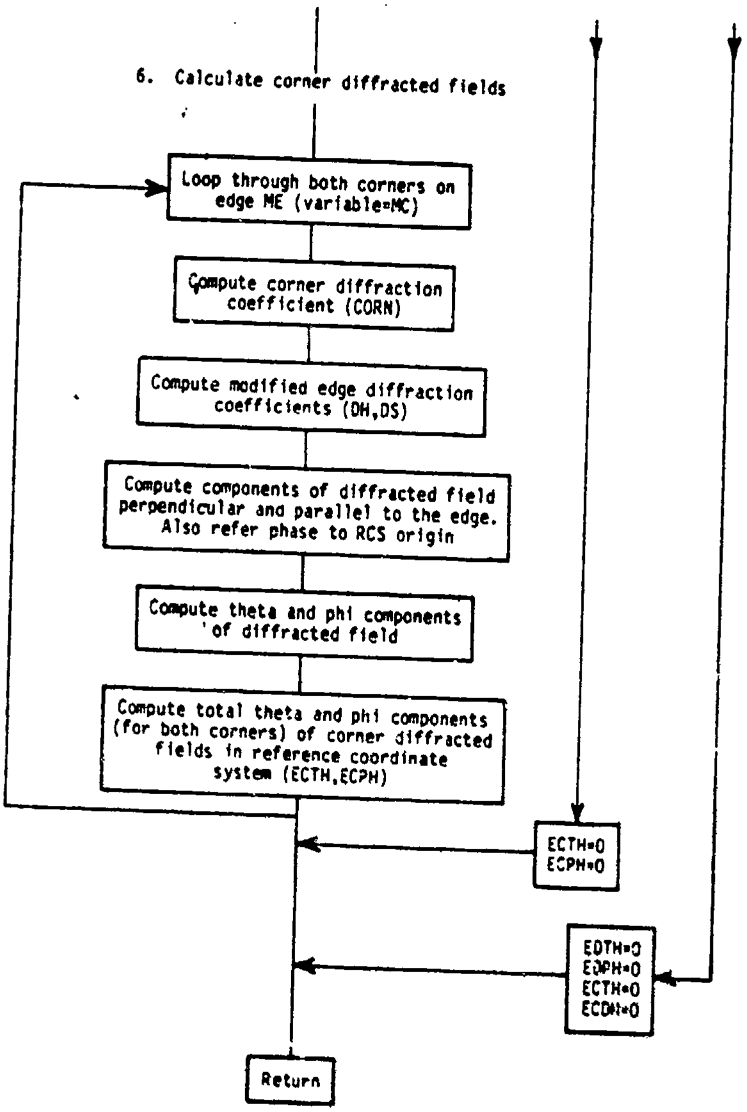
OUTPUT VARIABLES

| | |
|---|---|
| EDTH | theta component of edge diffracted E field |
| EDPH | phi component of edge diffracted E field |
| ECTH | theta component of corner diffracted E field |
| ECPH | phi component of corner diffracted E field |

1. Specify single reflection source image location
2. Perform diffraction point geometry calculations
Determine permissible range for diffraction angle
Determine if diffraction exists
No
Yes
Compute edge diffraction point and incident ray unit vector $\hat{VI}$
Is diffraction point on edge ME? If not, set at appropriate corner and set LDIF false
3. Check to see if ray is shadowed
Does diffracted ray hit another plate or a cylinder?
Yes
No


Does reflection from plate MR occur?
No
Yes
Does reflected ray hit another plate or a cylinder before diffraction?
Yes
No
Compute incident ray propagation direction on plate MR knowing reflected ray direction
Does incident ray from source hit a plate or a cylinder before reflection?
Yes
No
4. Calculate diffraction angles and related geometry
Calculate PSO and PS, the incident and diffracted phi angles
Compute diffraction polarization unit vectors (PHO,PH,BOP,BO) for incident and diffracted ray
Compute SBO=sine($B_o$)


5. Compute diffracted fields
Compute source pattern factor
Compute components of incident field perpendicular and parallel to the edge
If slope diffraction is desired compute incident slope field pattern factor
Compute phase term
Compute edge diffraction coefficients
Compute components of edge diffracted field perpendicular and parallel to the edge
Does edge diffraction exist? (LDIF=TRUE)
No
Yes
EDTH=0
EDPH=0
Compute theta and phi components of edge diffracted field in RCS (EDTH,EDPH)
Is corner diffraction desired?
No
Yes


6. Calculate corner diffracted fields
Loop through both corners on edge ME (variable=MC)
Compute corner diffraction coefficient (CORN)
Compute modified edge diffraction coefficients (DH,DS)
Compute components of diffracted field perpendicular and parallel to the edge. Also refer phase to RCS origin
Compute theta and phi components of diffracted field
Compute total theta and phi components (for both corners) of corner diffracted fields in reference coordinate system (ECTH,ECPH)
ECTH=0
ECPH=0
EDTH=0
EDPH=0
ECTH=0
ECPH=0
Return

## SYMBOL DICTIONARY

| | |
|---|---|
| ADN | DOT PRODUCT OF VECTOR FROM PLATE MP TO THE SOURCE IMAGE AND THE PLATE UNIT NORMAL |
| AFN | WEDGE ANGLE NUMBER |
| BDEL | VARIABLE USED TO EXPAND DIFFRACTION ANGLE RANGE IF CORNER DIFFRACTION IS USED |
| BDHI | UPPER LIMIT FOR BD, THE COSINE OF THE DIFFRACTION ANGLE BETA |
| BDLOW | LOWER LIMIT FOR BD, THE COSINE OF THE DIFFRACTION ANGLE BETA |
| BETN | DIFFERENCE IN DIFFRACTED AND INCIDENT PHI ANGLES |
| BETP | SUM OF DIFFRACTED AND INCIDENT PHI ANGLES |
| BO | DIFFRACTED FIELD BETA POLARIZATION UNIT VECTOR IN DIFFRACTION EDGE COORDINATE SYSTEM (IN X,Y,Z RCS COMPONENTS) |
| BOP | INCIDENT FIELD BETA POLARIZATION UNIT VECTOR IN DIFFRACTION EDGE COORDINATE SYSTEM (IN X,Y,Z RCS COMPONENTS) |
| BRD | LOWER AND UPPER LIMIT FOR EDGE DIFFRACTION ANGLE<br>BRD(1)=COS(BLOW)<br>BRD(2)=COS(BHIGH) |
| CNP | COSINE OF HALF WEDGE ANGLE |
| CORN | CORNER DIFFRACTION COEFFICIENT |
| CPH | COSINE OF PSR |
| CPHJ | COSINE OF PHJR |
| CPHO | COSINE OF PSOR |
| CTH | COSINE OF THR |
| CTHJ | COSINE OF THJR |
| CTHP | COSINE OF THPR |
| DEL | PARAMETER USED IN TRANSITION FUNCTION |
| DH | DIFFRACTION COEF. FOR HARD BOUNDARY CONDITION |
| DHIR | DISTANCE FROM REFLECTION POINT TO DIFFRACTION POINT |
| DHIT | DISTANCE FROM SOURCE TO REFLECTION POINT (FROM PLAINT) |
| DHT | DISTANCE FROM SOURCE TO HIT (FROM PLAINT AND CYLINT) |
| DIN | EDGE DIFFRACTION COEFFICIENT (FROM SUB. DI) FOR INCIDENT DIFFRACTED FIELD |
| DIP | EDGE DIFFRACTION COEFFICIENT (FROM SUB. DI) FOR REFLECTED DIFFRACTED FIELD |
| DPH | SLOPE DIFFRACTION COEFFICIENT FOR HARD BOUNDARY CONDITION |
| DPS | SLOPE DIFFRACTION COEFFICIENT FOR SOFT BOUNDARY CONDITION |
| DS | DIFFRACTION COEF. FOR SOFT BOUNDARY CONDITION |
| DV | DOT PRODUCT OF EDGE UNIT VECTOR AND DIFFRACTED RAY PROPAGATION DIRECTION UNIT VECTOR |
| ECPH | PHI COMPONENT OF CORNER DIFFRACTED E-FIELD |
| ECTH | THETA COMPONENT OF CORNER DIFFRACTED E-FIELD |
| EDPH | PHI COMPONENT OF EDGE DIFFRACTED E-FIELD |
| EDPL | COMPONENT OF DIFFRACTED FIELD PARALLEL TO THE EDGE |
| EDPR | COMPONENT OF DIFFRACTED FIELD PERPENDICULAR TO THE EDGE |
| EDTH | THETA COMPONENT OF EDGE DIFFRACTED E-FIELD |
| EF | THETA COMPONENT OF CORNER DIFFRACTED E-FIELD IN RCS |
| EG | PHI COMPONENT OF CORNER DIFFRACTED E-FIELD IN RCS |
| EIPL | COMPONENT OF INCIDENT FIELD PARALLEL TO THE EDGE |
| EIPLP | PATTERN FACTOR FOR COMPONENT OF INCIDENT SLOPE FIELD PARALLEL TO THE EDGE |
| EIPR | COMPONENT OF INCIDENT FIELD PERPENDICULAR TO THE EDGE |
| EIPRP | PATTERN FACTOR FOR COMPONENT OF INCIDENT SLOPE FIELD PERPENDICULAR TO THE EDGE |
| EIX | |
| EIY | SOURCE PATTERN FACTORS FOR X,Y, AND Z COMPONENTS OF INCIDENT |
| EIZ | E FIELD |
| EXPH | COMPLEX PHASE TERM (REFER PHASE TO RCS. ORIGIN) |
| FN | WEDGE ANGLE NUMBER |
| FNN | WEDGE ANGLE INDICATOR |
| FNP | ANGLE EXTERIOR TO WEDGE IN DEGREES |
| GAM | DOT PRODUCT OF THE DIF RAY DIRECTION AND THE VECTOR FROM THE REF COORD SYS ORIGIN TO THE DIFFRACTION POINT |
| ISG | SIGN CHANGE VARIABLE |
| J | INDEX VARIABLE |
| LHIT | SET TRUE IF RAY HITS A PLATE OR CYLINDER (FROM PLAINT OR CYLINT) |
| MC | INDEX VARIABLE USED TO STEP THRU CORNERS |

| | |
|---|---|
| ME | EDGE ON PLATE MP WHERE DIFFRACTION OCCURS |
| MEC | CORNER AT END OF EDGE ME |
| MP | PLATE FOR WHICH DIFFRACTION OCCURS |
| MR | PLATE WHERE REFLECTION OCCURS |
| N | DO LOOP VARIABLE |
| NI | DO LOOOP VRIABLE |
| NJ | DO LOOP VARIABLE |
| PD | DOT PRODUCT OF DIF EDGE BINORMAL AND DIF RAY PROPAGATION DIRECTION |
| PH | DIFFRACTED FIELD PHI POLARIZATION UNIT VECTOR IN DIFFRACTION EDGE-FIXED COORDINATE SYSTEM (IN X,Y,Z RCS COMPONENTS) |
| PHIR | PHI COMPONENT OF REFL RAY PROPAGATION DIRECTION IN REF COORD SYS. |
| PHJR | PHI COMPONENT OF INCIDENT (SOURCE) RAY PROPAGATION DIRECTION |
| PHC | INCIDENT FIELD PHI POLARIZATION UNIT VECTOR IN DIFFRACTION EDGE-FIXED COORDINATE SYSTEM (IN X',Y,Z RCS COMPONENTS) |
| PHSR | PHI COMPONENT OF RAY PROPAGATION DIRECTION AFTER DIFFRACTION IN RCS |
| PP | NEGATIVE DOT PRODUCT OF DIF EDGE BINORMAL AND INCIDENT RAY UNIT VECTOR |
| PS | PSR*DPR |
| PSD | DIFFRACTED RAY PHI ANGLE IN EDGE-FIXED COORDINATE SYSTEM |
| PSO | PSOR*DPR |
| PSCD | INCIDENT RAY PHI ANGLE IN EDGE-FIXED COORDINATE SYSTEM |
| PSCR | PHI COMPONENT OF INCIDENT RAY DIRECTION IN EDGE FIXED COORDINATE SYSTEM |
| PSR | PHI COMPONENT OF DIFFRACTED RAY PROPAGATION DIRECTION IN EDGE-FIXED COORDINATE SYSTEM |
| QD | DOT PRODUCT OF DIF PLATE NORMAL AND DIF RAY PROPAGATION DIRECTION |
| QI | NEGATIVE OF DOT PRODUCT OF DIF PLATE NORMAL AND INCIDENT RAY PROPAGATION DIRECTION |
| SBO | SINE OF BO, THE ANGLE THE DIFFRACTED RAY MAKES WITH THE EDGE |
| SNP | SINE OF HALF WEDGE ANGLE |
| SP | DISTANCE FROM SOURCE IMAGE TO DIFFRACTION POINT (FROM SUB. DFPTND |
| SPH | SINE OF PSR |
| SPHJ | SINE OF PHJR |
| SPHO | SINE OF PSOR |
| SPP | DISTANCE FROM SOURCE IMAGE TO MODIFIED DIFFRACTION POINT |
| STHJ | SINE OF THJR |
| STHR | SINE OF THR |
| TERR | COEFFICIENT OF CORNER DIFFRACTED FIELDS |
| THIR | THETA COMPONENT OF REFLECTED RAY DIRECTION IN REF COORD SYS |
| THJR | THETA COMPONENT OF INCIDENT (SOURCE) RAY PROPAGATION DIRECTION |
| THPR | ANGLE DIFFRACTED RAY MAKES WITH EDGE |
| THR | ANGLE BETWEEN EDGE UNIT VECTOR AND RAY FROM SOURCE IMAGE LOCATION TO CORNER MC |
| TPP | DISTANCE PARAMETER USED IN CALCULATING DIFFRACTION COEFFICIENTS |
| VAX | 3X3 MATRIX DEFINING THE SOURCE IMAGE COORD SYS. AXES |
| VC | UNIT VECTOR FROM SOURCE IMAGE TO CORNER 1 OR 2 OF EDGE ME |
| VCM | DISTANCE FROM SOURCE IMAGE TO CORNER 1 OR 2 OF EDGE ME |
| VECT | VECTOR USED TO MOVE DIFFRACTION POINT OFF EDGE FOR SHADOWING TESTS |
| VI | UNIT VECTOR OF RAY INCIDENT ON EDGE FROM PLATE REFLECTION (FROM SUB. DFPTND) |
| VIP | UNIT VECTOR OF RAY FROM SOURCE IMAGE TO MODIFIED DIF POINT |
| VJ | X,Y, AND Z COMPONENTS OF SOURCE RAY PROPAGATION DIRECTION |
| VMC | DISTANCE ALONG THE EDGE FROM FIRST CORNER OF EDGE TO DIFFRACTION POINT |
| XP | DIFFRACTION POINT (CALCULATED IN SUB. DFPTND) IN RCS |
| XDP | MODIFIED DIFFRACTION POINT USED FOR SHADOWING TESTS |
| XIS | SOURCE IMAGE LOCATION (FOR REFLECTION FROM PLATE MR) |
| XS | SOURCE LOCATION IN REF COORD SYS |
| ZP | DOP PRODUCT OF PROPAGATION DIRECTION UNIT VECTOR AND VECTOR FROM DIFFRACTION POINT TO CORNER MC |

# CODE LISTING

```
C-----------------------------------------------------------------------
      SUBROUTINE RPLDPL(EDTH,EDPH,ECTH,ECPH,FNN,ME,MP,MR)
C!!!
C!!!  DETERMINES THE REFLECTED/DIFFRACTED FIELD WITH PHASE
C!!!  REFERRED TO ORIGIN. RAY IS REFLECTED FROM PLATE #MR AND
C!!!  DIFFRACTED FROM EDGE #ME ON PLATE #MP.
C!!!
      COMPLEX EF,EG,EIPR,EIPL,EXPH,DIN,DIP,EDPR,EDPL,EDTH,EDPH
      COMPLEX EIPRP,EIPLP,EIX,EIY,EIZ,CORN,FFCT
      COMPLEX DH,DS,DPH,DPS,ECBI,ECBR,ECTH,ECPH
      DIMENSION VI(3),XD(3),PHO(3),PH(3),BOP(3),BO(3),XDP(3)
      DIMENSION XIS(3),VJ(3),VC(2,3),VCM(2),BRD(2),VT(3),VIP(3)
      DIMENSION VAX(3,3)
      LOGICAL LHIT,LSURF
      LOGICAL LDEBUG,LTEST,LSLOPE,LCORNR,LDIF
      COMMON/TEST/LDEBUG,LTEST
      COMMON/LOGDIF/LSLOPE,LCORNR
      COMMON/EDMAG/VMAG(14,6)
      COMMON/GEOPLA/X(14,6,3),V(14,6,3),VP(14,6,3),VN(14,3)
     2,MEP(14),MPX
      COMMON/SORINF/XS(3),VXS(3,3)
      COMMON/IMAINF/XI(14,14,3),VXI(3,3,14)
      COMMON/DIR/D(3),THSR,PHSR,SPHS,CPHS,STHS,CTHS
      COMMON/THPHUV/DT(3),DP(2)
      COMMON/PIS/PI,TPI,DPR,RPD
      COMMON/SURFAC/LSURF(14)
      FN=FNN
C!!!  INITIALIZE FIELDS
      EDTH=(Ø.,Ø.)
      EDPH=(Ø.,Ø.)
      ECTH=(Ø.,Ø.)
      ECPH=(Ø.,Ø.)
      IF (LDEBUG) WRITE (6,106)
  106 FORMAT (/,' DEBUGGING RPLDPL SUBROUTINE')
      MEC=ME+1
      IF(MEC.GT.MEP(MP)) MEC=1
      DV=Ø.
      DO 10 N=1,3
      DV=DV+D(N)*V(MP,ME,N)
      IF(ABS(DV).GT.Ø.999) GO TO 40
C!!!  1.  SPECIFY SINGLE REFL CTION SOURCE IMAGE LOCATION
   10 XIS(N)=XI(MR,MR,N)
C!!!  2. PERFORM DIFFRACTION POINT GEOMETRY CALCULATIONS
C!!!  DETERMINE PERMISSABLE RANGE FOR DIFFRACTION ANGLE
      VCM(1)=Ø.
      VCM(2)=Ø.
      BRD(1)=Ø.
      BRD(2)=Ø.
      DO 11 N=1,3
      VC(1,N)=X(MP,ME,N)-XI(MR,MR,N)
      VC(2,N)=X(MP,MEC,N)-XI(MR,MR,N)
      VCM(1)=VCM(1)+VC(1,N)*VC(1,N)
   11 VCM(2)=VCM(2)+VC(2,N)*VC(2,N)
      VCM(1)=SQRT(VCM(1))
      VCM(2)=SQRT(VCM(2))
      DO 12 J=1,2
      DO 12 N=1,3
      VC(J,N)=VC(J,N)/VCM(J)
   12 BRD(J)=BRD(J)+V(MP,ME,N)*VC(J,N)
      BDEL=Ø.
      IF (LCORNR) BDEL=Ø.3
      BDLOW=BRD(1)-BDEL
      BDHI=BRD(2)+BDEL
C!!!  DETERMINE IF DIFFRACTION EXISTS
      IF(DV.LT.BDLOW.OR.DV.GT.BDHI) GO TO 40
C!!!  COMPUTE EDGE DIFFRACTION POINT AND INC RAY UNIT VECTOR VI
```

```
      CALL DFPTWD(XIS,DV,VI,SP,XD,ME,MP)
      VMG=0.
      ADN=0.
      AFN=FNN
      IF(AFN.GT.2.)AFN=6.-AFN
      CNP=COS(AFN*PI/2.)
      SNP=SIN(AFN*PI/2.)
      DO 15 N=1,3
      XDP(N)=XD(N)
      VMG=VMG+(XD(N)-X(MP,ME,N))*V(MP,ME,N)
   15 ADN=ADN+(XI(MR,MR,N)-X(MP,1,N))*VN(MP,N)
      LDIF=.TRUE.
C!!!  IS DIF POINT ON EDGE ME?
C!!!  IF NOT, SET AT APPROPRIATE CORNER AND SET LDIF FALSE
      IF (VMG.LT.0.) GO TO 101
      IF (VMG.LT.VMAG(MP,ME)-1.E-4) GO TO 102
      DO 103 N=1,3
  103 XDP(N)=X(MP,MEC,N)-1.E-4*V(MP,ME,N)
      LDIF=.FALSE.
      GO TO 102
  101 DO 104 N=1,3
  104 XDP(N)=X(MP,ME,N)+1.E-4*V(MP,ME,N)
      LDIF=.FALSE.
  102 DO 16 N=1,3
      VECT=VP(MP,ME,N)*CNP+VN(MP,N)*SNP
   16 XDP(N)=XDP(N)+1.E-5*VECT
C!!!  3.  CHECK TO SEE IF RAY IS SHADOWED
C!!!  DOES DIFFRACTED RAY HIT ANOTHER PLATE?
      CALL PLAINT(XDP,D,DHT,MP,LHIT)
      IF(LHIT) GO TO 40
C!!!  DOES DIFFRACTED RAY HIT A CYLINDER?
      CALL CYLINT(XDP,D,PHSR,DHT,LHIT,.TRUE.)
      IF(LHIT) GO TO 40
      SPP=0.
      DO 111 N=1,3
      VIP(N)=XDP(N)-XIS(N)
  111 SPP=SPP+VIP(N)*VIP(N)
      SPP=SQRT(SPP)
      DO 112 N=1,3
  112 VIP(N)=VIP(N)/SPP
C!!!  DOES REFLECTION FROM PLATE MR OCCUR?
      CALL PLAINT(XIS,VIP,DHIT,-MR,LHIT)
      IF (.NOT.LHIT) GO TO 40
      IF(DHIT.GT.SPP)GO TO 40
      DHIR=SPP-DHIT
      DHIR=DHIR-1.E-3
C!!!  DOES REFLECTED RAY HIT ANOTHER PLATE BEFORE DIFFRACTION?
      CALL PLAINT(XIS,VIP,DHT,MR,LHIT)
      IF(LHIT.AND.(DHT.LT.DHIR)) GO TO 40
      THIR=BTAN2(SQRT(VI(1)*VI(1)+VI(2)*VI(2)),VI(3))
      PHIR=BTAN2(VI(2),VI(1))
C!!!  DOES REFLECTED RAY HIT A CYLINDER.
      CALL CYLINT(XIS,VI,PHIR,DHT,LHIT,.TRUE.)
      IF(LHIT.AND.(DHT.LT.DHIR)) GO TO 40
C!!!  COMPUTE INCIDENT RAY DIRECTION ON PLATE #MR
C!!!  KNOWING REFLECTED DIRECTION.
      CALL REFBP(PHJR,THJR,PHIR,THIR,MR)
      SPHJ=SIN(PHJR)
      CPHJ=COS(PHJR)
      STHJ=SIN(THJR)
      CTHJ=COS(THJR)
      VJ(1)=CPHJ*STHJ
      VJ(2)=SPHJ*STHJ
      VJ(3)=CTHJ
C!!!  DOES INCIDENT RAY FROM SOURCE HIT A PLATE BEFORE REFLECTION?
      CALL PLAINT(XIS,VJ,DHT,MR,LHIT)
      IF(LHIT.AND.(DHT.LT.DHIT)) GO TO 40
C!!!  DOES INCIDENT RAY HIT A CYLINDER?
      CALL CYLINT(XS,VJ,PHJR,DHT,LHIT,.FALSE.)
      IF(LHIT.AND.(DHT.LT.DHIT)) GO TO 40
C!!!  4.  CALCULATE DIF ANGLES AND RELATED GEOMETRY
      QI=0.
      PP=0.
      QD=0.
      PD=0.
      DO 20 N=1,3
      QI=QI-VN(MP,N)*VI(N)
      PP=PP-VP(MP,ME,N)*VI(N)
      QD=QD+VN(MP,N)*D(N)
   20 PD=PD+VP(MP,ME,N)*D(N)
C!!!  CALCULATE PSO AND PS, THE INCIDENT AND DIF PHI ANGLES
      PSOR=BTAN2(QI,PP)
      PSO=DPR*PSOR
      IF(PSO.LT.0.) PSO=360.+PSO
      PSR=BTAN2(QD,PD)
      PS=DPR*PSR
      IF(PS.LT.0.) PS=360.+PS
      PSOD=PSO
      PSD=PS
      IF(FNN.LE.2.)GO TO 21
      FN=FNN-2.
      PSOD=360.-PSO
      PSD=360.-PS
   21 FNP=FN*180.+1.E-4
      IF(PSO.GT.FNP.OR.PS.GT.FNP) GO TO 40
      SPHO=SIN(PSOR)
      CPHO=COS(PSOR)
      SPH=SIN(PSR)
      CPH=COS(PSR)
C!!!  COMPUTE DIFFRACTION POLARIZATION UNIT
C!!!  VECTORS (PHO,PH,BOP,BO)
      DO 30 N=1,3
      PHO(N)=-VP(MP,ME,N)*SPHO+VN(MP,N)*CPHO
   30 PH(N)=-VP(MP,ME,N)*SPH+VN(MP,N)*CPH
      BOP(1)=PHO(2)*VI(3)-PHO(3)*VI(2)
      BOP(2)=PHO(3)*VI(1)-PHO(1)*VI(3)
      BOP(3)=PHO(1)*VI(2)-PHO(2)*VI(1)
      BO(1)=PH(2)*D(3)-PH(3)*D(2)
      BO(2)=PH(3)*D(1)-PH(1)*D(3)
      BO(3)=PH(1)*D(2)-PH(2)*D(1)
C!!!  COMPUTE SBO=SINE(BO)
      SBO=SQRT((V(MP,ME,3)*D(2)-V(MP,ME,2)*D(3))**2+(V(MP,ME,1)
     2*D(3)-V(MP,ME,3)*D(1))**2+(V(MP,ME,2)*D(1)-V(MP,ME,1)*D(2))
     2**2)
      TPP=SP*SBO*SBO
C!!!  5.  COMPUTE DIFFRACTED FIELDS
C!!!  COMPUTE SOURCE PATTERN FACTORS
      DO 29 NJ=1,3
      DO 29 NI=1,3
   29 VAX(NI,NJ)=VII(NI,NJ,MR)
      CALL SOURCE(EP,EQ,EIX,EIY,EIZ,THIR,PHIR,VAX)
C!!!  COMPUTE COMPONENTS OF INCIDENT FIELD PERP AND PARALLEL
C!!!  TO THE EDGE
      EIPR=EIX*PHO(1)+EIY*PHO(2)+EIZ*PHO(3)
      EIPL=EIX*BOP(1)+EIY*BOP(2)+EIZ*BOP(3)
C!!!  IF SLOPE DIF IS DESIRED, CALCULATE INCIDENT SLOPE FIELD
C!!!  PATTERN FACTORS
      IF(LSLOPE)CALL SOURCP(EIPRP,EIPLP,VI,PHO,BOP,VAX)
C!!!  COMPUTE PHASE TERM
      GAR=XD(1)*D(1)+XD(2)*D(2)+XD(3)*D(3)
      EIPH=CEXP(CMPLX(0.,TPI*(GAR-SP)))/SQRT(SP)
C!!!  COMPUTE EDGE DIFFRACTION COEFFICIENTS
```

```
      CALL DH (DS,DH,DPS,DPH,TPP,PSD,PSOD,SBO,FN,LSURF(MP))
      IF (LDEBUG) WRITE (6,*) EIPR,EIPL
      IF (LDEBUG) WRITE (6,*) DS,DH,DPS,DPH
      IF (LDEBUG) WRITE (6,*) TPP,PSD,PSOD,SBO,FN
C!!!  COMPUTE COMPENENTS OF EDGE DIF FIELD PERP. AND PARALLEL
C!!!  TO THE EDGE
      EDPR=-EIPR*DH*EXPH
      EDPL=-EIPL*DS*EXPH
      IF(.NOT.LSLOPE)GO TO 201
      EDPR=EDPR-EIPRP*DPH*EXPH/CMPLX(0.,TPI*SP*SBO)
      EDPL=EDPL-EIPLP*DPS*EXPH/CMPLX(0.,TPI*SP*SBO)
201   IF (.NOT.LDIF) GO TO 202
C!!!  COMPUTE THETA AND PHI COMPONENTS OF EDGE DIFF. FIELD
C!!!  IF DIFFRACTION EXISTS
      EDTH=EDPL*(BO(1)*DT(1)+BO(2)*DT(2)+BO(3)*DT(3))
     2+EDPR*(PH(1)*DT(1)+PH(2)*DT(2)+PH(3)*DT(3))
      EDPH=EDPL*(BO(1)*DP(1)+BO(2)*DP(2))
     2+EDPR*(PH(1)*DP(1)+PH(2)*DP(2))
C!!!  6.  IF CORNER DIF IS DESIRED, CALC CORNER DIF FIELDS
 202  IF (.NOT.LCORNR) GO TO 40
      BETN=PSD-PSOD
      BETP=PSD+PSOD
      EF=(0.,0.)
      EG=(0.,0.)
      MC=ME-1
      ISN=1
      J=0
C!!!  LOOP THRU BOTH CORNERS ON EDGE #ME
35    MC=MC+1
      IF(MC.GT.MEP(MP)) MC=1
      J=J+1
      ISN=-ISN
      CTH=-ISN*BRD(J)
      CTHP=ISN*DV
      THPR=ACOS(CTHP)
      THR=ACOS(CTH)
      STHR=SIN(THR)
      DEL=2.*TPI*VCM(J)*(COS(.5*(THR+THPR))**2)
      ZP=(X(MP,MC,1)-XD(1))*D(1)+(X(MP,MC,2)-XD(2))*D(2)
     2+(X(MP,MC,3)-XD(3))*D(3)
      TERM=-STHR/TPI/(CTH+CTHP)/SQRT(VCM(J))
C!!!  COMPUTE CORNER DIFFRACTION COEFFICIENT (CORN)
      CORN=-TERM*FFCT(DEL)*CEXP(CMPLX(0.,-TPI*(VCM(J)-SP-ZP)-.25*PI))
      CALL DI(DIN,TPP,BETN,SBO,FN,DEL,.TRUE.)
      IF(LSURF(MP))GO TO 311
      CALL DI(DIP,TPP,BETP,SBO,FN,DEL,.TRUE.)
C!!!  COMPUTE MODIFIED EDGE DIFF. COEFFICIENTS (DH,DS)
      DH=DIN+DIP
      DS=DIN-DIP
      GO TO 312
311   DH=DIN
      DS=(0.,0.)
C!!!  COMPUTE COMPONENTS OF DIF FIELD PERP. AND PARALLEL TO EDGE
312   EDPR=-EIPR*DH*EXPH
      EDPL=-EIPL*DS*EXPH
      IF(.NOT.LSLOPE)GO TO 203
      EDPR=EDPR-EIPRP*DPH*EXPH/CMPLX(0.,TPI*SP*SBO)
      EDPL=EDPL-EIPLP*DPS*EXPH/CMPLX(0.,TPI*SP*SBO)
C!!!  COMPUTE THET AND PHI COMPONENTS OF CORNER DIF FIELD
203   ECTH=EDPL*(BO(1)*DT(1)+BO(2)*DT(2)+BO(3)*DT(3))
     2+EDPR*(PH(1)*DT(1)+PH(2)*DT(2)+PH(3)*DT(3))
      ECPH=EDPL*(BO(1)*DP(1)+BO(2)*DP(2))
     2+EDPR*(PH(1)*DP(1)+PH(2)*DP(2))
C!!!  COMPUTE TOTAL THETA AND PHI COMPONENTS (FOR BOTH CORNERS)
C!!!  OF CORNER DIF FIELDS IN REF COORD SYS.
      EF=EF+ECTH*CORN
```

```
          EG=EG+ECPH*CORN
          IF (.NOT.LDEBUG) GO TO 36
          WRITE (6,*) DS,DH,EDPR,EDPL
          WRITE (6,*) ECTH,ECPH,CORN
          WRITE (6,*) EF,EG
    36    CONTINUE
          IF(MC.EQ.ME) GO TO 35
          ECTH=EF
          ECPH=EG
          RETURN
     40   IF (.NOT.LTEST) GO TO 204
          WRITE (6,205)
     205  FORMAT (/,' TESTING RPLDPL SUBROUTINE')
          WRITE (6,*) EDTH,EDPH,ECTH,ECPH
          WRITE (6,*) FN,ME,MP,MR
     204  RETURN
          END
```

RPLRCL

PURPOSE

To compute the geometricai optics field reflected by a given plate and then reflected by the cylinder.

PERTINENT GEOMETRY

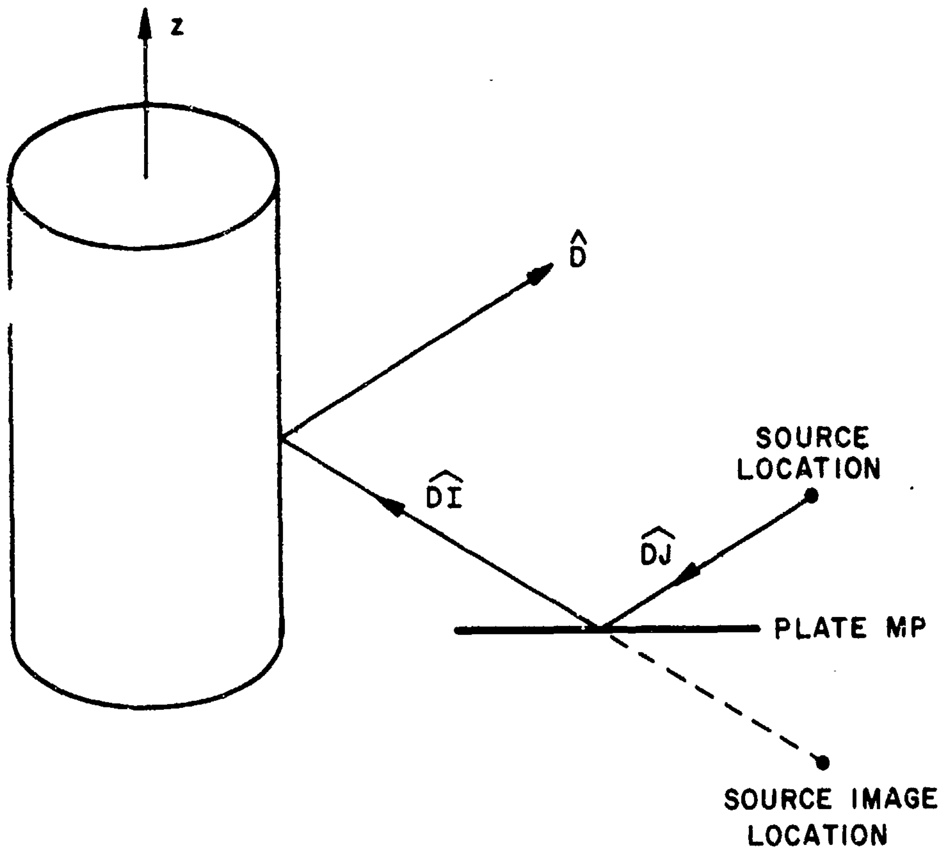


Figure 99--Illustration of plate reflected, cylinder reflected ray.

METHOD

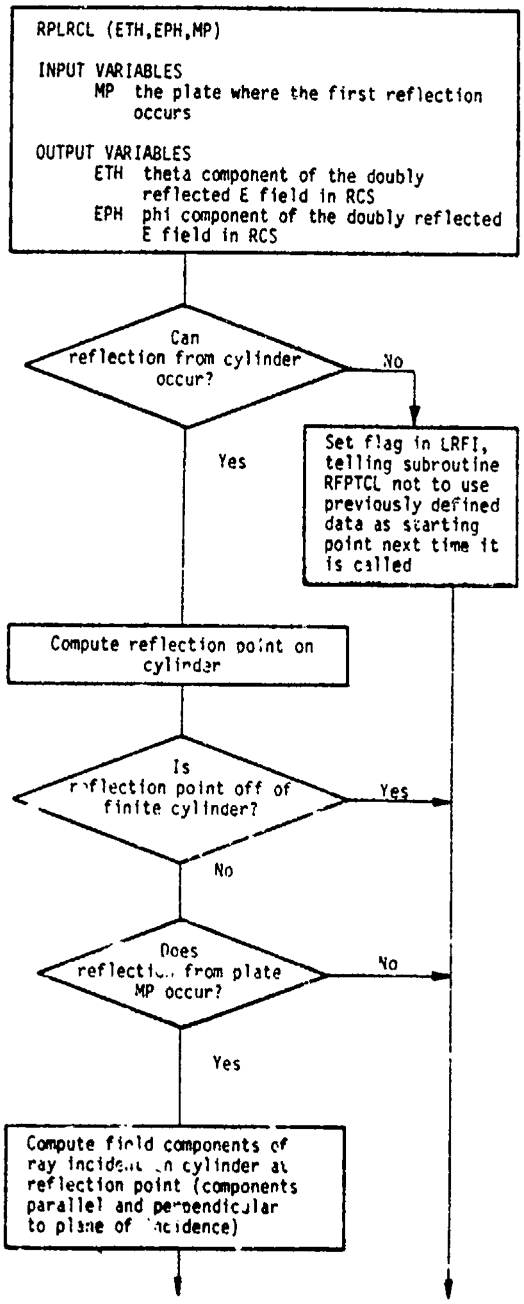
Subroutine RPLRCL functions as a service routine for subroutine RPLSCL, where the actual plate-cylinder fields are computed. The geometrical optics reflected field components ETH and EPH computed in RPLRCL are used only for reference purposes (when LOUT is set true). The field components calculated in RPLRCL which are used in RPLSCL, are the hard and soft components of the plate reflected field incident on the cylinder at the reflection point. These components, along with several other useful parameters, are passed to subroutine RPLSCL through common block FUDGI.

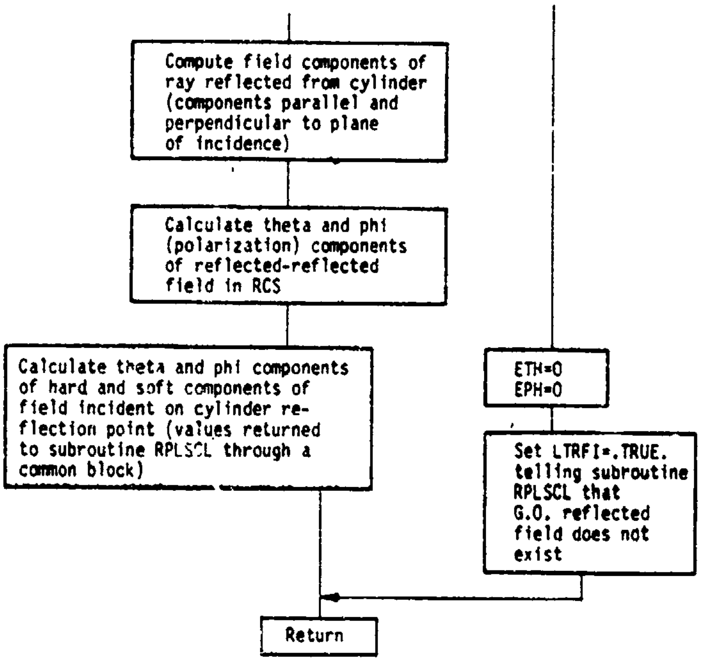
The geometrical optics fields determined in this subroutine for the reflection from the cylinder, are found in a similar manner to the fields calculated in subroutine REFCYL. However, in this subroutine the field incident on the cylinder is found from the image source for the particular plate of interest, as illustrated in Figure 99. The image source fields are calculated in a similar manner to those obtained in subroutine REFPLA. The phase of the resultant double reflected field is referred to the reference coordinate system origin. The double reflected field thus has the form

$$\overline{E}^{r,r} = W_m \, (ETH\hat{\theta}+EPH\hat{\phi}) \, \frac{e^{-jkR}}{R}$$

where the factor $\frac{e^{-jkR}}{R}$ and the source weight ($W_m$) are added elsewhere in the code.

## FLOW DIAGRAM


RPLRCL (ETH,EPH,MP)
INPUT VARIABLES
MP the plate where the first reflection occurs
OUTPUT VARIABLES
ETH theta component of the doubly reflected E field in RCS
EPH phi component of the doubly reflected E field in RCS
Can reflection from cylinder occur?
No
Set flag in LRFI, telling subroutine RFPTCL not to use previously defined data as starting point next time it is called
Yes
Compute reflection point on cylinder
Is reflection point off of finite cylinder?
Yes
No
Does reflection from plate MP occur?
No
Yes
Compute field components of ray incident on cylinder at reflection point (components parallel and perpendicular to plane of incidence)


Compute field components of ray reflected from cylinder (components parallel and perpendicular to plane of incidence)
Calculate theta and phi (polarization) components of reflected-reflected field in RCS
Calculate theta and phi components of hard and soft components of field incident on cylinder reflection point (values returned to subroutine RPLSCL through a common block)
ETH=0
EPH=0
Set LTRFI=.TRUE. telling subroutine RPLSCL that G.O. reflected field does not exist
Return

## SYMBOL DICTIONARY

| | |
|---|---|
| CTHW | DOT PRODUCT OF CYLINDER NORMAL AND REFLECTION PROPAGATION DIRECTION UNIT VECTOR |
| CW | COSINE OF WR |
| D | PROPAGATION DIRECTION AFTER CYL REFL. IN (X,Y,Z) RCS COMPONENTS |
| DD1 | DOT PRODUCT OF UNIT VECTOR OF PROPAGATION DIRECTION AND CYLINDER TANGENT UNIT VECTOR THROUGH TAN POINT 1 (2-D) |
| DD2 | DOT PRODUCT OF UNIT VECTOR OF PROPAGATION DIRECTION AND CYLINDER TANGENT UNIT VECTOR THROUGH TAN POINT 2 (2-D) |
| DHIS | DISTANCE FROM REFLECTION POINT ON PLATE TO REFLECTION POINT ON THE CYLINDER |
| DHIT | DISTANCE FROM SOURCE TO HIT POINT (FROM PLAINT) |
| DI | X,Y, AND Z COMPONENTS OF INCIDENT RAY DIRECTION ON CYL IN RCS |
| DJ | X,Y,Z COMPONENTS OF PROPAGATION DIRECTION OF RAY INCIDENT ON PLATE |
| DXY | DOT PRODUCT OF VECTOR FROM ORIGIN TO SOURCE IMAGE LOCATION AND PROPAGATION DIRECTION (2-D) |
| EF | PATTERN FACTOR OF THETA COMPONENT OF INCIDENT FIELD IN RCS |
| EG | PATTERN FACTOR OF PHI COMPONENT OF INCIDENT FIELD IN RCS |
| EHPH | PHI COMPONENT OF THE HARD COMPONENT OF FIELD INCIDENT ON CYL (PARALLEL TO PLANE OF INCIDENCE) |
| EHTH | THETA COMPONENT OF THE HARD COMPONENT OF FIELD INCIDENT ON CYL (PARALLEL TO PLANE OF INCIDENCE) |
| EIPP | INCIDENT CYL FIELD COMPONENT PARALLEL TO PLANE OF INCIDENCE |
| EIPR | INCIDENT CYL FIELD COMPONENT PERPENDICULAR TO PLANE OF INC |
| EPH | PHI COMPONENT OF CYL REFLECTED E-FIELD |
| ERPP | CYL REFLECTED FIELD COMPONENT PARALLEL TO PLANE OF INCIDENCE |
| ERPR | CYL REFLECTED FIELD COMPONENT PERPENDICULAR TO PLANE OF INC. |
| ERX<br>ERY<br>ERZ | X,Y,Z COMPONENTS OF FIELD INCIDENT ON (OR REFLECTED FROM) CYLINDER IN RCS |
| ESPH | PHI COMPONENT OF THE SOFT COMPONENT OF FIELD INCIDENT ON CYL (PERPENDICULAR TO PLANE OF INCIDENCE) |
| ESTH | THETA COMPONENT OF THE SOFT COMPONENT OF FIELD INCIDENT ON CYL (PERPENDICULAR TO PLANE OF INCIDENCE) |
| ETH | THETA COMPONENT OF CYL REFLECTED E FIELD |
| EX<br>EY<br>EZ | PATTERN FACTOR OF X,Y,Z COMPONENTS OF SOURCE FIELD INCIDENT ON CYLINDER IN RCS |
| LHIT | SET TRUE IF RAY HITS PLATE (FROM PLAINT) |
| LRFI | SET TRUE IF REFL DATA IS AVAILABLE FROM PREVIOUS PATTERN ANGLE (OR FOR NEXT PATTERN ANGLE (WHEN LEAVING ROUTINE)) |
| LTRFI | SET TRUE IF GEOMETRICAL OPTICS REFLECTED-REFLECTED FIELD DOES NOT EXIST |
| PH | COMPLEX PHASE AND RAY SPREADING COEFFICIENT |
| PHIR | PHI COMPONENT OF INCIDENT RAY DIRECTION ON CYL |
| RHO1 | RAY SPREADING RADIUS IN PLANE OF CYLINDER CURVATURE AT REFLECTION POINT |
| RHO2 | RAY SPREADING RADIUS NORMAL TO PLANE OF INCIDENCE AT REFLECTION POINT |
| SMAG | LENGTH OF RAY FROM REFL POINT ON CYL TO SOURCE IMAGE |
| SQRH | PART OF SPREADING FACTOR |
| SXN<br>SYN<br>SZN | X,Y, AND Z COMPONENTS OF UNIT VECTOR OF RAY FROM REFL. POINT ON CYLINDER TO SOURCE IMAGE LOCATION IN RCS |
| THIR | THETA COMPONENT OF INCIDENT RAY DIRECTION ON CYL |
| UIPPX<br>UIPPY<br>UIPPZ | X,Y,Z COMPONENTS OF INCIDENT FIELD POLARIZATION UNIT VECTOR PARALLEL TO PLANE OF INCIDENCE |
| UIPRX<br>UIPRY<br>UIPRZ | X,Y,Z COMPONENTS OF INC/REFL FIELD POLARIZATION UNIT VECTOR PERPENDICULAR TO PLANE OF INCIDENCE |
| URPPX<br>URPPY<br>URPPZ | X,Y,Z COMPONENTS OF REFL FIELD POLARIZATION UNIT VECTOR PARALLEL TO PLANE OF INCIDENCE |
| VAX | MATRIX DEFINING SOURCE COORDINATE SYS AXES IN RCS COMPONENTS |

```
XIS     X,Y,Z COMPONENTS OF SOURCE IMAGE LOCATION
        ALSO REFLECTION POINT ON PLATE
XR      LOCATION OF REFLECTION POINT ON CYL IN (X,Y,Z) RCS
```

## CODE LISTING

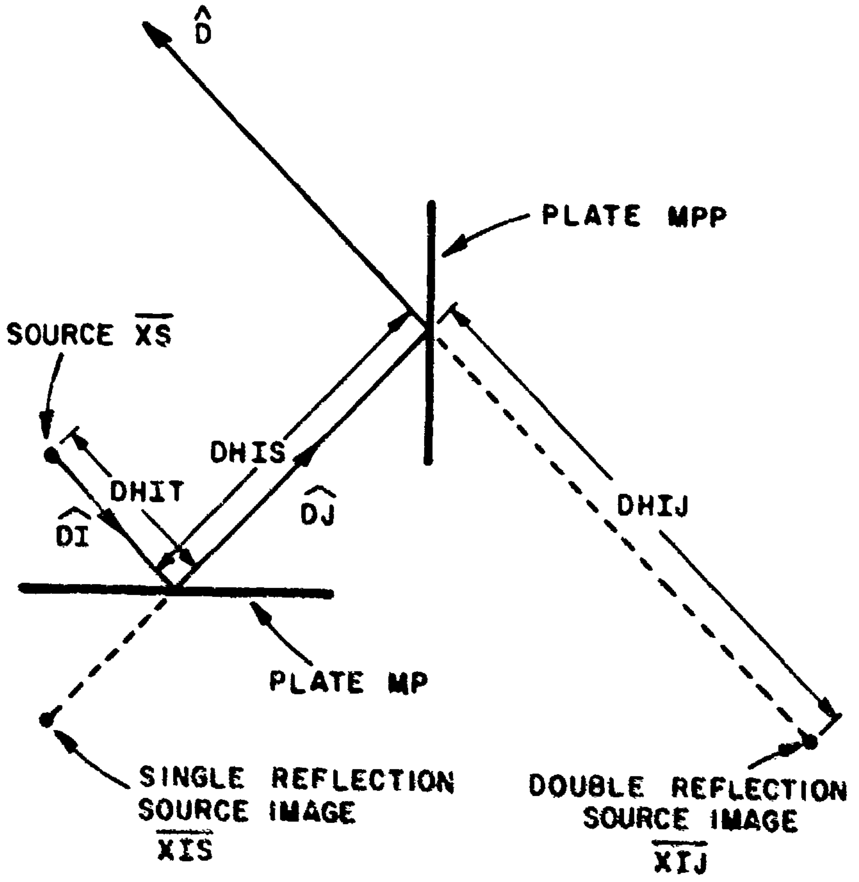
```
C-----------------------------------------------------------------------
      SUBROUTINE RPLRCL(ETH,EPH,MP)
C!!!
C!!!  COMPUTES THE G.O. FIELD REFLECTED FROM PLATE #MP THEN
C!!!  REFLECTED FROM THE ELLIPTIC CYLINDER
C!!!
      DIMENSION UN(2),UB(2),DI(3),DJ(3),XIS(3),VAX(3,3)
      COMPLEX ETH,EPH,EX,EY,EZ,PH,EIPR,EIPP,ERX,ERY,ERZ,ERPR,ERPP
      COMPLEX ESTH,ESPH,EHTH,EHPH,TRAN,EF,EG
      LOGICAL LHIT,LRFI,LDEBUG,LTEST,LTRFI
      COMMON/FUDGI/TRAN,ESTH,ESPH,EHTH,EHPH,XR(3),RG,RHO1,SMAG,LTRFI
      COMMON/GEOMEL/A,B,ZC(2),SNC(2),CNC(2),CTC(2)
      COMMON/SORINF/XS(3),VXS(3,3)
      COMMON/IMAINF/XI(14,14,3),VXI(3,3,14)
      COMMON/GEOPLA/X(14,6,3),V(14,6,3),VP(14,6,3),VN(14,3)
     2,MEP(14),MPX
      COMMON/PIS/PI,TPI,DPR,RPD
      COMMON/DIR/D(3),THSR,PHSR,SPS,CPS,STHS,CTHS
      COMMON/THPHUV/DT(3),DP(2)
      COMMON/ENDICL/DTI(14),VTI(14,2),BTI(14,4)
      COMMON/TEST/LDEBUG,LTEST
      COMMON/CLRFI/LRFI(14)
      IF(LDEBUG) WRITE(6,900)
  900 FORMAT(/,' DEBUGGING RPLRCL SUBROUTINE')
      LTRFI=.FALSE.
C!!!  CAN REFLECTION FROM CYLINDER OCCUR?
      IF(DTI(MP).LT.-1.5) GO TO 12
      DXY=XI(MP,MP,1)*CPS+XI(MP,MP,2)*SPS
      IF(DXY.GT.0.) GO TO 10
      DD1=BTI(MP,1)*CPS+BTI(MP,2)*SPS
      DD2=BTI(MP,3)*CPS+BTI(MP,4)*SPS
      IF(DD1.GT.DTI(MP).AND.DD2.GT.DTI(MP)) GO TO 12
   10 CONTINUE
C!!!  CALCULATE REFLECTION POINT ON CYLINDER
      CALL RFPTCL(PHSR,MP,VR,DOTP,DD,S,LRFI(MP))
      IF(DOTP.LE.0.) GO TO 11
      XR(1)=A*COS(VR)
      XR(2)=B*SIN(VR)
      XR(3)=XI(MP,MP,3)+S*CTHS/STHS
C!!!  IS REFLECTION POINT OFF OF FINITE CYLINDER?
      IF(XR(3).GT.ZC(1)+XR(1)*CTC(1).OR.
     2XR(3).LT.ZC(2)+XR(1)*CTC(2)) GO TO 11
C!!!  DOES CYLINDER REFLECTED RAY HIT A PLATE?
      CALL PLAINT(XR,D,DHT,0,LHIT)
      IF(LHIT) GO TO 11
      SXN=XI(MP,MP,1)-XR(1)
      SYN=XI(MP,MP,2)-XR(2)
      SZN=-S*CTHS/STHS
      SMAG=SQRT(SXN*SXN+SYN*SYN+SZN*SZN)
      SXN=SXN/SMAG
      SYN=SYN/SMAG
      SZN=SZN/SMAG
      PHIR=ATAN2(-SYN,-SXN)
      THIR=ATAN2(SQRT(SXN*SXN+SYN*SYN),-SZN)
      SPHI=SIN(PHIR)
      CPHI=COS(PHIR)
      STHI=SIN(THIR)
      CTHI=COS(THIR)
      DI(1)=CPHI*STHI
      DI(2)=SPHI*STHI
      DI(3)=CTHI
      DO 15 N=1,3
   15 XIS(N)=XI(MP,MP,N)
C!!!  DOES REFLECTION OFF OF PLATE MP OCCUR?
      CALL PLAINT(XIS,DI,DHIT,-MP,LHIT)
      IF(.NOT.LHIT) GO TO 11
      DHIS=SMAG-DHIT
C!!!  DOES REFLECTED RAY HIT PLATE BEFORE THE CYLINDER?
      CALL PLAINT(XIS,DI,DHT,MP,LHIT)
      IF(LHIT.AND.(DHT.LT.DHIS)) GO TO 11
      CALL REFBP(PHJR,THJR,PHIR,THIR,MP)
      SPHJ=SIN(PHJR)
      CPHJ=COS(PHJR)
      STHJ=SIN(THJR)
      CTHJ=COS(THJR)
      DJ(1)=CPHJ*STHJ
      DJ(2)=SPHJ*STHJ
      DJ(3)=CTHJ
C!!!  DOES SOURCE RAY INC. ON PLATE MP HIT ANOTHER PLATE
C!!!  OR THE CYLINDER FIRST?
      CALL PLAINT(XS,DJ,DHT,MP,LHIT)
      IF(LHIT.AND.(DHT.LT.DHIT)) GO TO 11
      CALL CYLINT(XS,DJ,PHJR,DHT,LHIT,.FALSE.)
      IF(LHIT.AND.(DHT.LT.DHIT)) GO TO 11
      DO 20 NJ=1,3
      DO 20 NI=1,3
20    VAX(NI,NJ)=VXI(NI,NJ,MP)
C!!!  CALCULATE SOURCE PATTERN FACTOR
      CALL SOURCE(EF,EG,EX,EY,EZ,THIR,PHIR,VAX)
      IF(LDEBUG) WRITE(6,*) EF,EG
      RG=DD*DD*DD/A/B
      CALL NANDB(UN,UB,VR)
      CTHW=UN(1)*D(1)+UN(2)*D(2)
      WR=BTAN2(SXN*UB(1)+SYN*UB(2),SZN)
      SW=SIN(WR)
      CW=COS(WR)
      SST2=SW*SW+CW*CW*CTHW*CTHW
      RHO2=SMAG
      RHO1=SMAG*RG*CTHW/(RG*CTHW+2.*SMAG*SST2)
C!!!  COMPUTE POLARIZATION UNIT VECTORS
C!!!  PERPENDICULAR AND PARALLEL TO PLANE OF INCIDENCE
      UIPRX=SIN(WR-PI/2.)*UB(1)
      UIPRY=SIN(WR-PI/2.)*UB(2)
      UIPRZ=COS(WR-PI/2.)
      UIPPX=SYN*UIPRZ-SZN*UIPRY
      UIPPY=SZN*UIPRX-SXN*UIPRZ
      UIPPZ=SXN*UIPRY-SYN*UIPRX
      URPPX=UIPRY*D(3)-UIPRZ*D(2)
      URPPY=UIPRZ*D(1)-UIPRX*D(3)
      URPPZ=UIPRX*D(2)-UIPRY*D(1)
      PH=CEXP(CMPLX(0.,-TPI*SMAG))/SMAG
C!!!  CALCULATE INCIDENT FIELD COMPONENTS PARALLEL
C!!!  AND PERPENDICULAR TO PLANE OF INCIDENCE
      EIPR=(UIPRX*EX+UIPRY*EY+UIPRZ*EZ)
      EIPP=(UIPPX*EX+UIPPY*EY+UIPPZ*EZ)
      PH=PH*CEXP(CMPLX(0.,TPI*(XR(1)*D(1)+XR(2)*D(2)+XR(3)*D(3))))
      SQRH=SQRT(RHO1*RHO2)
C!!!  COMPUTE REFLECTED FIELD COMPONENTS PARALLEL
C!!!  AND PERPENDICULAR TO PLANE OF INCIDENCE
      ERPR=-SQRH*PH*EIPR
      ERPP=SQRH*PH*EIPP
      TRAN=SQRH*PH
      ERX=ERPR*UIPRX+ERPP*URPPX
      ERY=ERPR*UIPRY+ERPP*URPPY
      ERZ=ERPR*UIPRZ+ERPP*URPPZ
C!!!  CALCULATE THETA AND PHI COMPONENTS OF REFLECTED-
C!!!  REFLECTED FIELD
      ETH=ERX*DT(1)+ERY*DT(2)+ERZ*DT(3)
      EPH=ERX*DP(1)+ERY*DP(2)
C!!!  COMPUTE THETA AND PHI COMPONENT OF SOFT COMPONENT OF
C!!!  FIELD INC. ON CYLINDER
      ERX=EIPR*UIPRX
      ERY=EIPH*UIPRY
      ERZ=EIPH*UIPRZ
      ESTH=ERX*DT(1)+ERY*DT(2)+ERZ*DT(3)
      ESPH=ERX*DP(1)+ERY*DP(2)
C!!!  COMPUTE THETA AND PHI COMPONENT OF HARD COMPONENT OF
C!!!  FIELD INC. ON CYLINDER
      ERX=EIPP*URPPX
      ERY=EIPP*URPPY
      ERZ=EIPP*URPPZ
      EHTH=ERX*DT(1)+ERY*DT(2)+ERZ*DT(3)
      EHPH=ERX*DP(1)+ERY*DP(2)
      GO TO 905
12    LRFI(NP)=.FALSE.
11    LTRFI=.TRUE.
      ETH=(0.,0.)
      EPH=(0.,0.)
905   CONTINUE
      IF(.NOT.LTEST) RETURN
      WRITE(6,910)
910   FORMAT(/,' TESTING RPLRCL SUBROUTINE')
      WRITE(6,*) ETH,EPH,NP
      RETURN
      END
```

## RPLRPL

PURPOSE

To calculate the far zone electric field due to double reflection from specified plates (reflection off of plate MP a̶ıd then plate MPP).

PERTINENT GEOMETRY


$\hat{D}$
PLATE MPP
SOURCE $\overline{XS}$
DHIS
DHIT
$\widehat{DI}$
$\widehat{DJ}$
DHIJ
PLATE MP
SINGLE REFLECTION SOURCE IMAGE $\overline{XIS}$
DOUBLE REFLECTION SOURCE IMAGE $\overline{XIJ}$

Figur. 100--Geometry for double reflected ray.

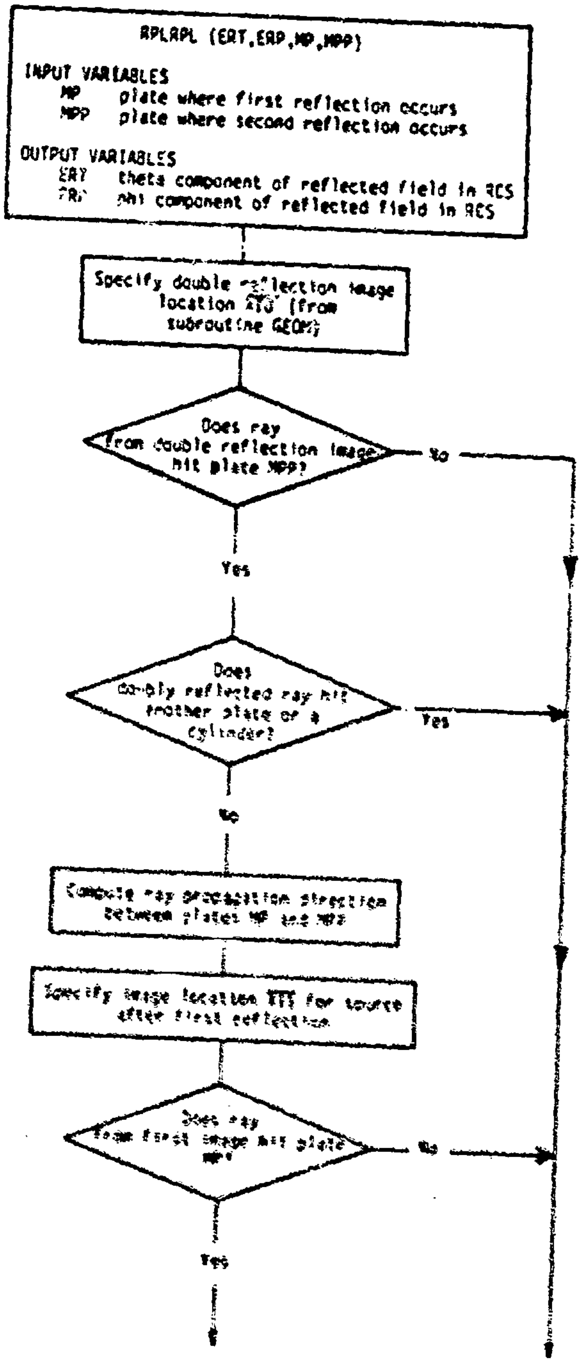
METHOD

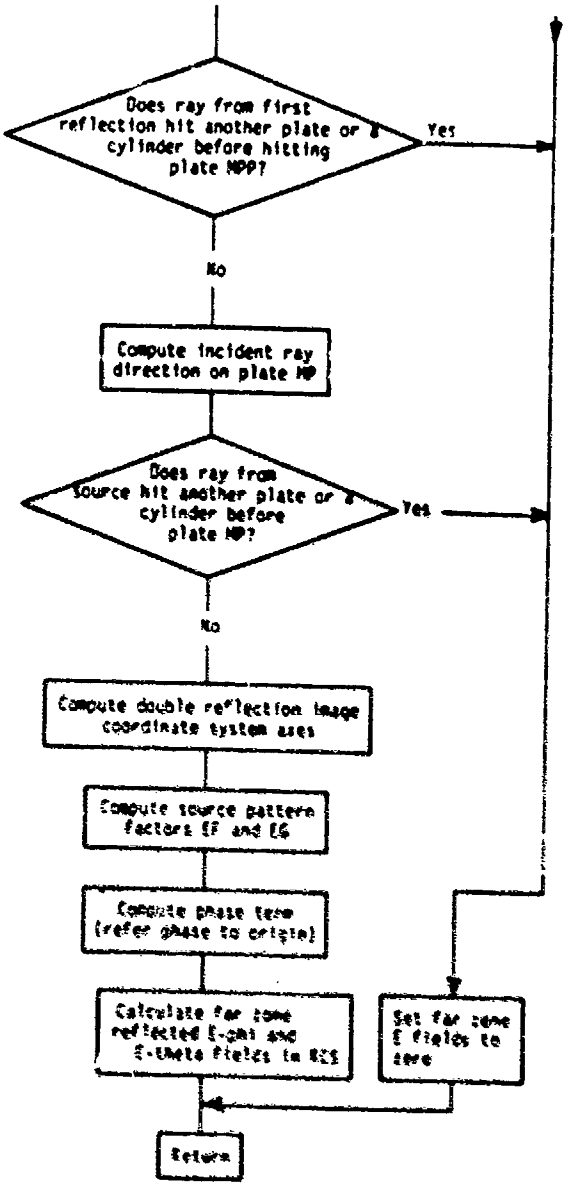
The doubly reflected fields are found using image theory. The double reflection source image is found so that the appropriate boundary conditions are met at the reflection points. The ray paths are checked to insure that they hit the appropriate plates and are not shadowed by other obstacles. The phase factor, $e^{jkD\cdot\overline{XIJ}}$, is then added to the pattern factor obtained from the SOURCE subroutine. The doubly reflected field is given in the form

$$\vec{E}^{rr}(r,\theta,\phi) = W_m(ERT\hat{\theta} + ERP\hat{\phi})\,\frac{e^{-jkR}}{R}\;.$$

The factor $\frac{e^{-jkR}}{R}$ and the source weight ($W_m$) are added elsewhere in the code.

## FLOW DIAGRAM


RPLRPL (ERT,ERP,MP,MPP)
INPUT VARIABLES
MP plate where first reflection occurs
MPP plate where second reflection occurs
OUTPUT VARIABLES
ERT theta component of reflected field in RCS
ERP phi component of reflected field in RCS
Specify double reflection image location XTU (from subroutine GEOM)
Does ray from double reflection image hit plate MPP?
No
Yes
Does doubly reflected ray hit another plate or a cylinder?
Yes
No
Compute ray propagation direction between plates MP and MPP
Specify image location XTT for source after first reflection
Does ray from first image hit plate MP?
No
Yes


Does ray from first reflection hit another plate or a cylinder before hitting plate MPP?
Yes
No
Compute incident ray direction on plate MP
Does ray from source hit another plate or a cylinder before plate MP?
Yes
No
Compute double reflection image coordinate system axes
Compute source pattern factors EF and EG
Compute phase term (refer phase to origin)
Calculate far zone reflected E-phi and E-theta fields in RCS
Set far zone E fields to zero
Return

## SYMBOL DICTIONARY

| | |
|---|---|
| CPHI | COSINE OF PHIR |
| CPHJ | COSINE OF PHJR |
| CPHS | COSINE OF PHSR |
| CTHI | COSINE OF THIR |
| CTHJ | COSINE OF THJR |
| CTHS | COSINE OF THSR |
| D | X,Y,Z COMPONENTS OF RAY PROPAGATION DIRECTION AFTER SECOND REFLECTION IN RCS |
| DHIJ | DISTANCE FROM DOUBLE REFLECTION IMAGE TO HIT POINT ON PLATE MPP |
| DHIS | DISTANCE BETWEEN REFLECTION POINTS |
| DHIT | DISTANCE FROM SOURCE TO REFLECTION POINT (FROM PLAINT) |
| DI | X,Y,Z COMPONENTS OF INCIDENT RAY PROPAGATION DIRECTION IN RCS |
| DJ | X,Y,Z COMPONENTS OF PROPAGATION DIRECTION OF RAY INCIDENT ON PLATE MPP |
| EX | COMPLEX PHASE FACTOR (CEXP(J*TPI*GAM)) |
| GAM | PHASE DISTANCE TO ORIGIN (DOT PRODUCT OF DOUBLE REFLECTION IMAGE LOCATION AND REFLECTED RAY PROPAGATION DIRECTION) |
| LHIT | SET TRUE IF RAY INTERSECTS A PLATE OR CYLINDER (FROM PLAINT OR CYLINT) |
| MP | PLATE FROM WHICH FIRST REFLECTION OCCURS |
| MPP | PLATE FROM WHICH SECOND REFLECTION OCCURS |
| PHIR | PHI COMPONENT OF INCIDENT RAY PROPAGATION DIRECTION IN RCS |
| PHJR | PHI COMPONENT OF RAY DIRECTION BETWEEN REFLECTIONS IN RCS |
| PHSR | PHI COMPONENT OF RAY PROPAGATION DIRECTION AFTER REFLECTION IN RCS |
| SPHI | SINE OF PHIR |
| SPHJ | SINE OF PHJR |
| SPHS | SINE OF PHSR |
| STHI | SINE OF THIR |
| STHJ | SINE OF THJR |
| THIR | THETA COMPONENT OF INCIDENT RAY PROPAGATION DIRECTION IN RCS |
| THJR | THETA COMPONENT OF RAY DIRECTION BETWEEN REFLECTIONS IN RCS |
| THSR | THETA COMPONENT OF RAY PROPAGATION DIRECTION AFTER REFLECTIONS IN RCS |
| VAX | X,Y,Z COMPONENTS DEFINING UNIT VECTORS OF THE SOURCE IMAGE COORDINATE SYSTEM AXES IN RCS COMPONENTS |
| VAXP | X,Y,Z COMPONENTS DEFINING UNIT VECTORS OF THE SOURCE IMAGE COORDINATE SYSTEM AXES IN RCS FOR DOUBLE REFLECTION |
| XI | TRIPLY DIMENSIONED ARRAY OF IMAGE LOCATIONS |
| XIJ | X,Y,Z COMPONENTS OF DOUBLE REFLECTION IMAGE LOCATION |
| XIS | X,Y,Z COMPONENTS OF SINGLE REFLECTION SOURCE IMAGE LOCATION (SINGLE REFLECTION FROM PLATE MP) |
| XS | SOURCE LOCATION IN (X,Y,Z) RCS |

CODE LISTING

```
C-------------------------------------------------------------------
      SUBROUTINE RPLRPL(ERT,ERP,MP,MPP)
C!!!
C!!!  DETERMINES THE REFL./REFL. FIELD WITH PHASE REFERRED TO
C!!!  ORIGIN. RAY IS REFL. BY PLATE#MP THEN BY PLATE#MPP.
C!!!
      COMPLEX EF,EG,EX,ERT,ERP,EIX,EIY,EIZ
      DIMENSION XIS(3),XIJ(3),DI(3),DJ(3),VAX(3,3),VAXP(3,3)
      LOGICAL LHIT
      LOGICAL LDEBUG,LTEST
      COMMON/TEST/LDEBUG,LTEST
      COMMON/DIR/D(3),THSR,PHSR,SPHS,CPHS,STHS,CTHS
      COMMON/GEOPLA/X(14,6,3),V(14,6,3),VP(14,6,3),VN(14,3)
     2,MEP(14),MPX
      COMMON/SORINF/XS(3),VXS(3,3)
      COMMON/IMAINF/XI(14,14,3),VXI(3,3,14)
      COMMON/PIS/PI,TPI,DPR,RPD
      IF (LDEBUG) WRITE (6,101)
 101  FORMAT (/,' DEBUGGING RPLRPL SUBROUTINE')
C!!!  SPECIFY IMAGE POSITION AFTER DOUBLE REFL.
      DO 5 N=1,3
 5    XIJ(N)=XI(MP,MPP,N)
C!!!  DOES RAY FROM DOUBLE REFL. IMAGE HIT PLATE #MPP
      CALL PLAINT(XIJ,D,DHIJ,-MPP,LHIT)
      IF(.NOT.LHIT) GO TO 50
C!!!  DOES DOUBLE REFL. RAY HIT ANOTHER PLATE
      CALL PLAINT(XIJ,D,DHT,MPP,LHIT)
      IF(LHIT) GO TO 50
C!!!  DOES DOUBLE REFL. RAY HIT A CYLINDER
      CALL CYLINT(XIJ,D,PHSR,DHT,LHIT,.TRUE.)
      IF(LHIT) GO TO 50
C!!!  COMPUTE THE RAY DIR BETWEEN PLATES MP AND MPP (DJ)
      CALL REFBP(PHJR,THJR,PHSR,THSR,MPP)
      IF (LDEBUG) WRITE (6,*) PHJR,THJR,PHSR,THSR,MPP
      SPHJ=SIN(PHJR)
      CPHJ=COS(PHJR)
      STHJ=SIN(THJR)
      CTHJ=COS(THJR)
      DJ(1)=CPHJ*STHJ
      DJ(2)=SPHJ*STHJ
      DJ(3)=CTHJ
C!!!  SPECIFY IMAGE LOCATION FOR SOURCE AFTER FIRST REFLECTION
      DO 6 N=1,3
 6    XIS(N)=XI(MP,MP,N)
C!!!  DOES RAY FROM FIRST IMAGE HIT PLATE #MP
      CALL PLAINT(XIS,DJ,DHIT,-MP,LHIT)
      IF(.NOT.LHIT) GO TO 50
      DHIS=DHIJ-DHIT
      DHIS=DHIS-1.E-3
C!!!  DOES RAY FROM FIRST IMAGE HIT ANOTHER PLATE BEFORE PLATE MPP?
      CALL PLAINT(XIS,DJ,DHT,MP,LHIT)
      IF(LHIT.AND.(DHT.LT.DHIS)) GO TO 50
C!!!  DOES RAY HIT A CYLINDER
      CALL CYLINT(XIS,DJ,PHJR,DHT,LHIT,.TRUE.)
      IF(LHIT.AND.(DHT.LT.DHIS)) GO TO 50
C!!!  KNOWING RAD. DIRECTION COMPUTE INCIDENT DIRECTION
C!!!  ON PLATE #MP
      CALL REFBP(PHIR,THIR,PHJR,THJR,MP)
      IF (LDEBUG) WRITE (6,*) PHIR,THIR,PHJR,THJR,MP
      SPHI=SIN(PHIR)
      CPHI=COS(PHIR)
      STHI=SIN(THIR)
      CTHI=COS(THIR)
      DI(1)=CPHI*STHI
      DI(2)=SPHI*STHI
      DI(3)=CTHI
```

```
C!!!  DOES RAY FROM SOURCE HIT ANOTHER PLATE BEFORE PLATE MP?
      CALL PLAINT(XS,DI,DHT,MP,LHIT)
      IF(LHIT.AND.(DHT.LT.DHIT)) GO TO 5Ø
C!!!  DOES RAY FROM SOURCE HIT A CYLINDER
      CALL CYLINT(XS,DI,PHIR,DHT,LHIT,.FALSE.)
      IF(LHIT.AND.(DHT.LT.DHIT)) GO TO 5Ø
C!!!  COMPUTE DOUBLE REFL. SOURCE IMAGE COORD SYS AXES
      DO 4Ø NJ=1,3
      DO 4Ø NI=1,3
4Ø    VAX(NI,NJ)=VXI(NI,NJ,MP)
      CALL IMDIR(VAXP,VAX,MPP)
C!!!  IF REFL/REFL FIELD EXISTS COMPUTE THE SOURCE PATTERN FACTORS
      CALL SOURCE(EF,EG,EIX,EIY,EIZ,THSR,PHSR,VAXP)
      IF (LDEBUG) WRITE (6,*) EF,EG
C!!!  COMPUTE PHASE REFERRED TO ORIGIN
      GAM=XI(MP,MPP,1)*D(1)+XI(MP,MPP,2)*D(2)+XI(MP,MPP,3)*D(3)
      EX=CEXP(CMPLX(Ø.,TPI*GAM))
C!!!  CALCULATE FAR-ZONE E-PHI AND E-THETA FIELDS
      ERT=EF*EX
      ERP=EG*EX
      GO TO 1
5Ø    CONTINUE
      ERT=(Ø.,Ø.)
      ERP=(Ø.,Ø.)
1     IF (.NOT.LTEST) GO TO 2
      WRITE (6,3)
3     FORMAT (/,' TESTING RPLRPL SUBROUTINE')
      WRITE (6,*) ERT,ERP,MP,MPP
2     RETURN
      END
```

## RPLSCL

PURPOSE

To calculate the far-zone electric field of a source ray which is reflected by a given plate and then scattered by the cylinder.

PERTINENT GEOMETRY

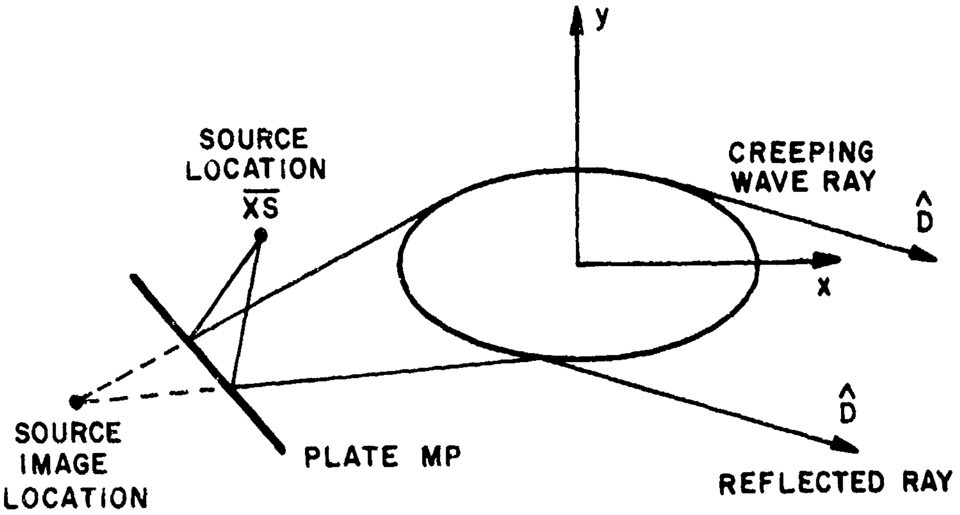


Figure 101--Illustration of ray reflected by a plate and then scattered by the cylinder.

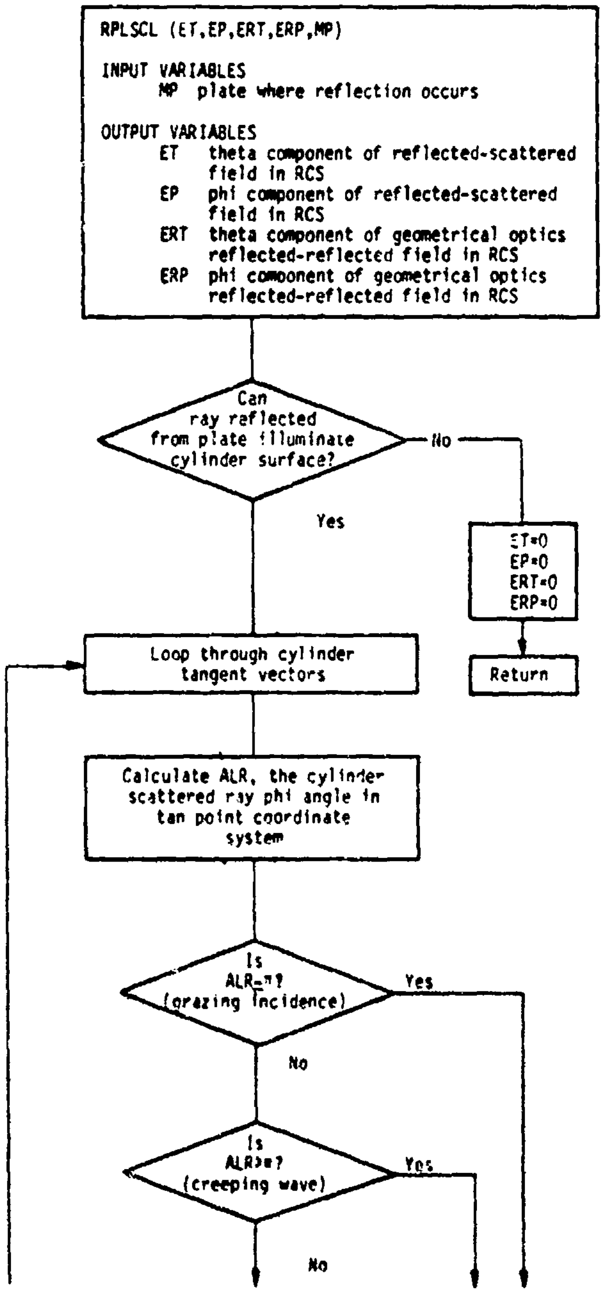
METHOD

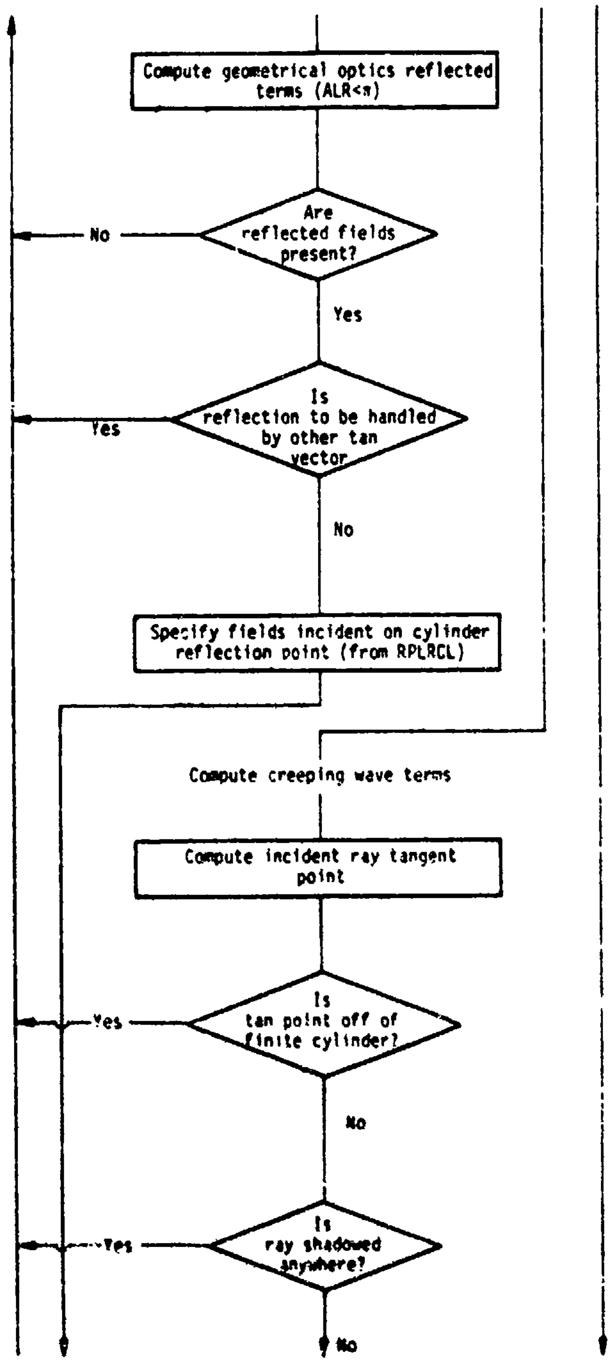
A uniform Geometrical Theory of Diffraction solution for the field reflected by a plate then reflected or diffracted by a cylinder is computed in this subroutine. The fields reflected or diffracted by the cylinder are determined in a similar manner as the fields calculated in subroutine SCTCYL. However, the incident field is found from the image source for the particular plate of interest, as illustrated in Figure 101. The image source fields are calculated in a similar manner to those obtained in subroutine REFPLA. The phase of the resultant reflected-scattered fields are referred to the reference coordinate system origin. The form of this field is then given by

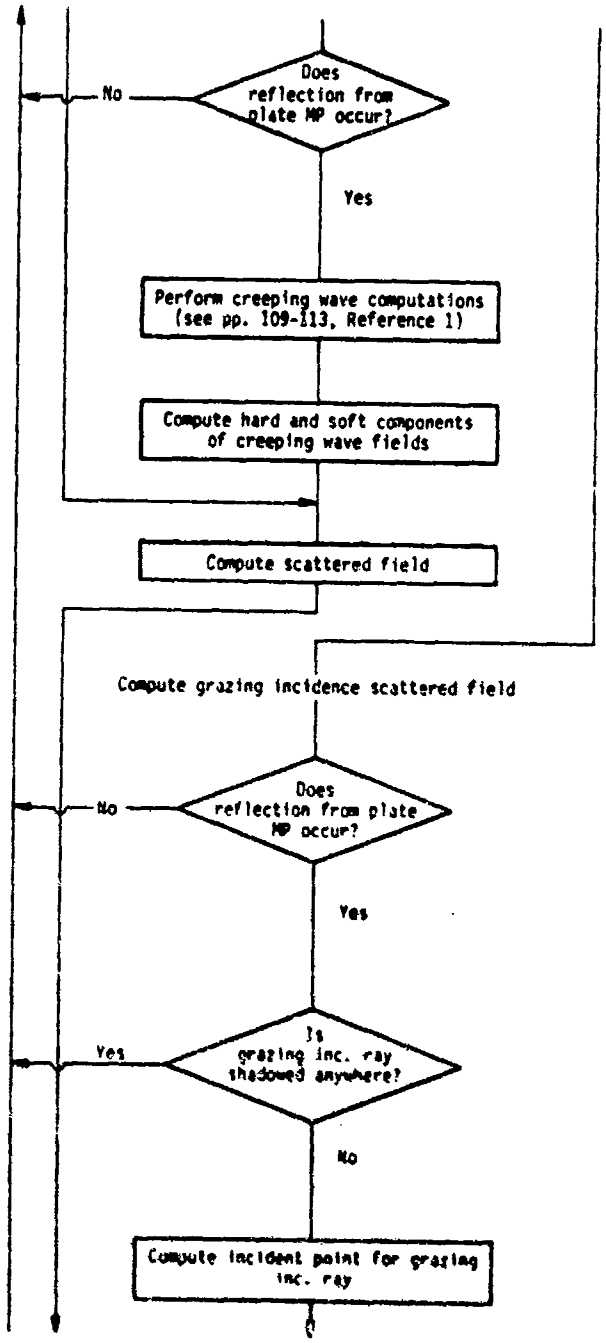
$$\bar{E}^{r,s} = W_m(ET\hat{\theta}+EP\hat{\phi})\,\frac{e^{-jkR}}{R}\ ,$$

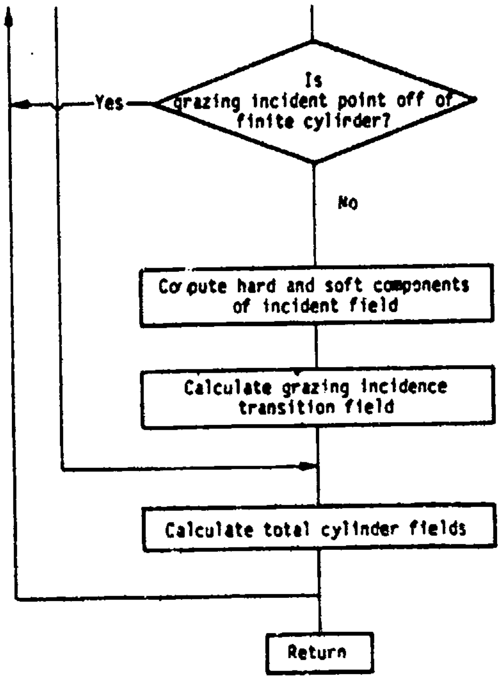
where the factor $\frac{e^{-jkR}}{R}$ and the source weight ($W_m$) are added elsewhere in the code.

FLOW DIAGRAM


RPLSCL (ET,EP,ERT,ERP,MP)
INPUT VARIABLES
MP plate where reflection occurs
OUTPUT VARIABLES
ET theta component of reflected-scattered field in RCS
EP phi component of reflected-scattered field in RCS
ERT theta component of geometrical optics reflected-reflected field in RCS
ERP phi component of geometrical optics reflected-reflected field in RCS
Can ray reflected from plate illuminate cylinder surface?
No
Yes
ET=0
EP=0
ERT=0
ERP=0
Return
Loop through cylinder tangent vectors
Calculate ALR, the cylinder scattered ray phi angle in tan point coordinate system
Is ALR=π? (grazing incidence)
Yes
No
Is ALR>π? (creeping wave)
Yes
No


Compute geometrical optics reflected terms (ALR<π)
No
Are reflected fields present?
Yes
Yes
Is reflection to be handled by other tan vector
No
Specify fields incident on cylinder reflection point (from RPLRCL)
Compute creeping wave terms
Compute incident ray tangent point
Yes
Is tan point off of finite cylinder?
No
Yes
Is ray shadowed anywhere?
No


Does reflection from plate MP occur?
No
Yes
Perform creeping wave computations (see pp. 109-113, Reference 1)
Compute hard and soft components of creeping wave fields
Compute scattered field
Compute grazing incidence scattered field
Does reflection from plate MP occur?
No
Yes
Is grazing inc. ray shadowed anywhere?
Yes
No
Compute incident point for grazing inc. ray


Is
grazing incident point off of
finite cylirder?
Yes
No
Co pute hard and soft components
of incident field
Calculate grazing incidence
transition field
Calculate total cylinder fields
Return

## SYMBOL DICTIONARY

| | |
|---|---|
| ALR | CYLINDER REFLECTED RAY PHI ANGLE IN TAN POINT COORDINATE SYSTEM (2-D) |
| ALS | PHI ANGLE DEFINING DIRECTION OF RAY FROM RCS ORIGIN TO SOURCE IMAGE IN TAN POINT COORD SYS |
| BX<br>BY<br>BZ | X,Y,Z COMPONENTS OF POLARIZATION UNIT VECTOR OF SOFT COMPONENT OF FIELD INCIDENT ON CYL (PARALLEL TO CYL SURFACE AND NORMAL TO INC RAY PROP DIR) |
| CFH | HARD TRANSITION FIELD COEFFICIENT |
| CFS | SOFT TRANSITION FIELD COEFFICIENT |
| DEPH | PHI COMPONENT OF TRANSITION FIELD IN RCS |
| DETH | THETA COMPONENT OF TRANSITION FIELD IN RCS |
| DHIT | DISTANCE FROM SOURCE IMAGE TO PLATE REFLECTION POINT (FROM PLAINT) |
| DHIV | DISTANCE FROM PLATE REFLECTION POINT TO CYLINDER |
| DHT | DISTANCE FROM SOURCE TO HIT POINT (FROM PLAINT) |
| DI | UNIT VECTOR OF RAY INCIDENT ON CYLINDER |
| DJ | X,Y,Z COMPONENTS OF UNIT VECTOR OF PROPAGATION DIRECTION OF SOURCE RAY INCIDENT ON PLATE |
| EF | PATTERN FACTOR FOR THETA COMPONENT OF INCIDENT FIELD IN RCS |
| EG | PATTERN FACTOR FOR PHI COMPONENT OF INCIDENT FIELD IN RCS |
| EHP | PHI COMPONENT OF HARD COMPONENT OF FIELD INCIDENT ON CYLINDER IN RCS |
| EHT | THETA COMPONENT OF HARD COMPONENT OF FIELD INCIDENT ON CYLINDER IN RCS |
| EP | PHI COMPONENT OF CYLINDER SCATTERED E FIELD WITH PHASE REFERRED TO RCS ORIGIN |
| ER | DOT PRODUCT OF UNIT VECTOR TANGENT TO CYLINDER AND THE PROPAGATION DIR. UNIT VECTOR |
| ESP | PHI COMPONENT OF SOFT COMPONENT OF FIELD INCIDENT ON CYLINDER IN RCS |
| EST | THETA COMPONENT OF SOFT COMPONENT OF FIELD INCIDENT ON CYLINDER IN RCS |
| ET | THETA COMPONENT OF CYLINDER SCATTERED E FIELD WITH PHASE REFERRED TO RCS ORIGIN |
| EX<br>EY<br>EZ | PATTERN FACTOR FOR X,Y,Z COMPONENTS OF INCIDENT FIELD IN RCS |
| I | VARIABLE USED TO STEP THROUGH TANGENT POINTS |
| LHIT | SET TRUE IF RAY HITS A PLATE (FROM PLAINT) |
| LTRFI | (RETURNED FROM RPLRCL) SET TRUE IF G.O. CYLINDER REFLECTED FIELD DOES NOT EXIST |
| PHIR | PHI COMPONENT OF PROPAGATION DIRECTION OF RAY INCIDENT ON CYLINDER |
| PHJR | PHI COMPONENT OF PROPAGATION DIRECTION OF SOURCE RAY INCIDENT ON PLATE |
| S | LENGTH OF VECTOR FROM SOURCE IMAGE TO TAN POINT (2 OR 3-D) |
| THIR | THETA COMPONENT OF PROPAGATION DIRECTION OF RAY INCIDENT ON CYLINDER |
| THJR | THETA COMPONENT OF PROPAGATION DIRECTION OF SOURCE RAY INCIDENT ON PLATE |
| VP | ELL ANGLE DEFINING POINT WHERE CREEPING WAVE LEAVES CYLINDER |
| VI | ELL. ANGLE USED TO DEFINE TANGENT POINTS (2-D) |
| VL | ELL ANGLE DEFINING LOWER RANGE OF CREEPING WAVE TRAVEL ON CYLINDER (2-D) |
| VU | ELL ANGLE DEFINING UPPER RANGE OF CREEPING WAVE TRAVEL ON CYLINDER (2-D) |
| XD<br>YD<br>ZD | X,Y,Z COMPONENTS OF DIRECTION OF RAY FROM SOURCE TO CYLINDER TANGENT POINT (INCIDENT RAY FOR CREEPING AND GRAZING INC. CASES) |

XII<br>
YII } X,Y,Z COMPONENTS OF POINT WHERE INCIDENT CREEPING<br>
ZII   WAVE (OR GRAZING WAVE) MEETS CYLINDER

XIS   X,Y,Z COMPONENTS OF IMAGE SOURCE LOCATION (FOR REFLECTION FROM PLATE MP)

XPP   X,Y,Z COMPONENTS OF POINT WHERE RAY LEAVES CYLINDER

XRF   X,Y,Z COMPONENTS OF POINT WHERE CREEPING WAVE LEAVES CYLINDER

## CODE LISTING

```
  C-----------------------------------------------------------------
        SUBROUTINE RPLSCL(ET,EP,ERT,ERP,MP)
  C!!!
  C!!!  COMPUTES THE FIELD REFLECTED FROM PLATE #MP THEN
  C!!!  SCATTERED FROM THE ELLIPTIC CYLINDER
  C!!!
        COMPLEX CJ,CPI4,CF,CFH,CFS,FI,PFUN,OFUN
        COMPLEX EIX,EIY,EIZ,EIPH,EITH,ET,EP,ERT,ERP
        COMPLEX REF,ESTH,ESPH,EHTH,EHPH,DETH,DEPH,EF,EG
        COMPLEX EST,ESP,EHT,EHP
        DIMENSION VI(2),ER(2),UN(2),UB(2),DI(3),XRF(3)
        DIMENSION XIS(3),DJ(3),VAX(3,3)
        LOGICAL LHIT,LTRFI,LDEBUG,LTEST,LRFI,LRFIT
        COMMON/GEOMEL/A,B,ZC(2),SNC(2),CNC(2),CTC(2)
        COMMON/SORINF/XS(3),VXS(3,3)
        COMMON/IMAINF/XI(14,14,3),VXI(3,3,14)
        COMMON/PIS/PI,TPI,DPR,RPD
        COMMON/CTD/AS,IO,SAS,SASP,CAS
        COMMON/DIR/D(3),THSR,PHSR,SPS,CPS,STHS,CTHS
        COMMON/COMP/CJ,CPI4
        COMMON/BNDICL/DTI(14),VTI(14,2),BTI(14,4)
        COMMON/FUDGI/REF,ESTH,ESPH,EHTH,EHPH,XR(3),RG,RHO1,DL,LTRFI
        COMMON/TEST/LDEBUG,LTEST
        COMMON/CLRFI/LRFI(14)
        EXTERNAL FCT
        IF(LDEBUG) WRITE(6,900)
  900   FORMAT(/,' DEBUGGING RPLSCL SUBROUTINE')
        ET=(0.,0.)
        EP=(0.,0.)
        ERT=(0.,0.)
        ERP=(0.,0.)
  C!!!  CAN PLATE REFLECTED RAY ILLUMINATE CURVED SURFACE?
        IF(DTI(MP).LT.-1.5) GO TO 909
        ER(1)=BTI(MP,1)*CPS+BTI(MP,2)*SPS
        ER(2)=BTI(MP,3)*CPS+BTI(MP,4)*SPS
  C!!!  LOOP THRU TANGENT VECTORS
        I=1
        LRFIT=.FALSE.
        VI(1)=VTI(MP,1)
        VI(2)=VTI(MP,2)
  3     CONTINUE
        CALL NANDB(UN,UB,VI(1))
        SINA=UN(1)*CPS+UN(2)*SPS
  C!!!  CALCULATE ALR, THE REFLECTED RAY PHI ANGLE IN
  C!!!  TANGENT POINT COORD. SYS.
        ALR=ATAN2(SINA,-ER(1))
        IF(ALR.LT.0.) ALR=ALR+TPI
  C!!!  IF GRAZING INCIDENCE IS PRESENT, SKIP TO
  C!!!  APPROPRIATE SECTION
        IF(ABS(PI-ALR).LT.0.0005) GO TO 5
  C!!!  IF ALR.GT.PI COMPUTE CREEPING WAVE TERMS
        IF(ALR.GT.PI) GO TO 10
  C!!!  COMPUTE REFLECTED FIELD TERMS IF ALR .LE. PI
        CALL RPLRCL(ERT,ERP,MP)
  C!!!  ARE REFLECTED FIELDS PRESENT?
        IF(LTRFI) GO TO 1
        SNAS=UN(1)*XI(MP,MP,1)+UN(2)*XI(MP,MP,2)
        IC=2*I-1
        CSAS=BTI(MP,IC)*XI(MP,MP,1)+BTI(MP,IC+1)*XI(MP,MP,2)
        ALS=ATAN2(SNAS,-CSAS)
        ALRS=ALR-ALS
  C!!!  IS REFLECTION TO BE HANDLED BY OTHER TANGENT VECTOR?
        IF(ABS(ALRS).LT.0.0005.AND.I.EQ.2) GO TO 1
        IF(ALRS.LE.-0.0005) GO TO 1
        GN=(PI*RG)**(1./3.)
        RHS=RHO1*DL/(DL-RHO1)
```

```
      SKWIG=-ABS(2.*TPI*RHS/GM/GM)
      CF=-SQRT(-2./PI/SKWIG)*CPI4*REF
      CF=CF*CEXP(-CJ*SKWIG*SKWIG*SKWIG/12.)
      TTRM=SKWIG/GM
      XX=PI*(DL+RHS)*TTRM*TTRM
C!!!  SPECIFY HARD AND SOFT COMPONENTS OF FIELD INC. ON CYLINDER
C!!!  FROM RPLRCL
      EST=ESTH
      ESP=ESPH
      EHT=EHTH
      EHP=EHPH
      GO TO 30
10    CONTINUE
      IF(LRFIT) LRFI(MP)=.FALSE.
      LRFIT=.TRUE.
C!!!  COMPUTE CREEPING WAVE TERMS IF ALR .GT. PI
C!!!  COMPUTE INCIDENT RAY TANGENT POINT
      XII=A*COS(VI(I))
      YII=B*SIN(VI(I))
      XD=XII-XI(MP,MP,1)
      YD=YII-XI(MP,MP,2)
      S=SQRT(XD*XD+YD*YD)
      ZII=S*CTHS/STHS+XI(MP,MP,3)
C!!!  IS TAN POINT ON (FINITE) CYLINDER?
      IF(ZII.GT.ZC(1)+XII*CTC(1).OR.
     2ZII.LT.ZC(2)+XII*CTC(2)) GO TO 1
      ZD=ZII-XI(MP,MP,3)
      PHIR=BTAN2(YD,XD)
      THIR=BTAN2(S,ZD)
      S=SQRT(S*S+ZD*ZD)
      DI(1)=XD/S
      DI(2)=YD/S
      DI(3)=ZD/S
      DO 15 N=1,3
15    XIS(N)=XI(MP,MP,N)
C!!!  DOES REFLECTION OFF OF PLATE MP OCCUR?
      CALL PLAINT(XIS,DI,DHIT,-MP,LHIT)
      IF(.NOT.LHIT) GO TO 1
      DHIV=S-DHIT
C!!!  IS RAY SHADOWED BETWEEN REFLECTION AND DIFFRACTION?
      CALL PLAINT(XIS,DI,DHT,MP,LHIT)
      IF(LHIT.AND.(DHT.LT.DHIV)) GO TO 1
C!!!  CALCULATE PROPAGATION DIRECTION OF RAY INCIDENT
C!!!  ON PLATE MP
      CALL REFBP(PHJR,THJR,PHIR,THIR,MP)
      SPHJ=SIN(PHJR)
      CPHJ=COS(PHJR)
      STHJ=SIN(THJR)
      CTHJ=COS(THJR)
      DJ(1)=CPHJ*STHJ
      DJ(2)=SPHJ*STHJ
      DJ(3)=CTHJ
C!!!  IS SOURCE RAY SHADOWED BEFORE HITTING PLATE MP?
      CALL PLAINT(XS,DJ,DHT,MP,LHIT)
      IF(LHIT.AND.(DHT.LT.DHIT)) GO TO 1
      CALL CYLINT(XS,DJ,PHJR,DHT,LHIT,.FALSE.)
      IF(LHIT.AND.(DHT.LT.DHIT)) GO TO 1
      DO 26 NJ=1,3
C!!!  SPECIFY SOURCE IMAGE AXES AND CALCULATE
C!!!  SOURCE PATTERN FACTOR
      DO 26 NI=1,3
26    VAX(NI,NJ)=VXI(NI,NJ,MP)
      CALL SOURCE(EF,EG,EIX,EIY,EIZ,THIR,PHIR,VAX)
C!!!  PERFORM CREEPING WAVE COMPUTATIONS
      IF(LDEBUG) WRITE(6,*) EF,EG
      IF(I.EQ.1) VD=BTAN2(-B*CPS,A*SPS)
      IF(I.EQ.2) VD=BTAN2(B*CPS,-A*SPS)
      VDP=VD-VI(I)
      IF(VDP.GT.PI) VDP=VDP-TPI
      IF(VDP.LT.-PI) VDP=VDP+TPI
      IF(I.EQ.2) GO TO 20
      IF(VDP.LT.0.) GO TO 1
      VL=VI(I)
      VU=VDP+VI(I)
      GO TO 25
20    CONTINUE
      IF(VDP.GT.0.) GO TO 1
      VL=VDP+VI(I)
      VU=VI(I)
25    CONTINUE
      CALL FKARG(SKNIG,AS,VL,VU)
      XRF(1)=A*COS(VD)
      XRF(2)=B*SIN(VD)
      ID=3
      CALL DOG32(VL,VU,FCT,SS)
      SS=SS/SAS
      XRF(3)=ZII+SS*CTHS
C!!!  DOES RAY HIT PLATE AFTER LEAVING CYLINDER?
      CALL PLAINT(XRF,D,DHIT,0,LHIT)
      IF(LHIT) GO TO 1
      CALL RADCV(RGI,RT,VI(I))
      CALL RADCV(RGF,RT,VD)
      GMM=(PI*PI*RGI*RGF)**(1./6.)
      CF=-GMM*CPI4*CEXP(-CJ*TPI*(S+SS))/PI/SQRT(2.*S)
      CF=CF*CEXP(CJ*TPI*(XRF(1)*D(1)+XRF(2)*D(2)+XRF(3)*D(3)))
      TTRM=SKNIG/GMM
      XX=PI*S*TTRM*TTRM
      BX=-UN(2)*DI(3)
      BY=UN(1)*DI(3)
      BZ=UN(2)*DI(1)-UN(1)*DI(2)
      ESP=(0.,0.)
      EHT=(0.,0.)
C!!!  COMPUTE HARD AND SOFT CREEPING WAVE COMPONENTS
      EHP=EIX*UN(1)+EIY*UN(2)
      EST=EIX*BX+EIY*BY+EIZ*BZ
      IF(I.EQ.1) EHP=-EHP
      IF(I.EQ.2) EST=-EST
30    CONTINUE
C!!!  COMPUTE THE SCATTERED FIELD
      XXS=SQRT(TPI*XX)
      XXX=SQRT(2.*XX/PI)
      CALL FRNELS(CCC,SSS,XXX)
      FI=CMPLX(0.5-CCC,SSS-0.5)
      FI=XXS*FI*CEXP(CJ*(.5*PI+XX))
      FI=-FI/SKNIG/SQRT(2.)
      SQTP=SQRT(2.*PI)
      CFH=CF*(FI-SQTP*GFUN(SKNIG))
      CFS=CF*(FI+SQTP*PFUN(SKNIG))
      DEPH=CFH*EHP+CFS*ESP
      DETH=CFH*EHT+CFS*EST
      GO TO 6
5     CONTINUE
C!!!  COMPUTE GRAZING INC. SCATTERED FIELD
      DO 35 N=1,3
35    XIS(N)=XI(NP,NP,N)
C!!!  DOES REFLECTION FROM PLATE NP OCCUR IN GRAZING INCIDENCE
C!!!  DIRECTION?
      CALL PLAINT(XIS,D,DHIT,-NP,LHIT)
      IF(.NOT.LHIT) GO TO 1
C!!!  IS RAY SHADOWED BETWEEN REFLECTIONS?
      CALL PLAINT(XIS,D,DHT,NP,LHIT)
      IF(LHIT) GO TO 1
      CALL REFBP(PHJR,THJR,PHSR,THSR,MP)
      SPHJ=SIN(PHJR)
      CPHJ=COS(PHJR)
      STHJ=SIN(THJR)
      CTHJ=COS(THJR)
      DJ(1)=CPHJ*STHJ
      DJ(2)=SPHJ*STHJ
      DJ(3)=CTHJ
C!!!  IS INCIDENT (SOURCE) RAY SHADOWED BY PLATE OR CYLINDER?
      CALL PLAINT(XS,DJ,DHT,MP,LHIT)
      IF(LHIT.AND.(DHT.LT.DHIT)) GO TO 1
      CALL CYLINT(XS,DJ,PHJR,DHT,LHIT,.FALSE.)
      IF(LHIT.AND.(DHT.LT.DHIT)) GO TO 1
C!!!  CALCULATE GRAZING INCIDENCE POINT
      SGN=-SIGN(1.,SIN(ALR))
      XII=A*COS(VI(I))
      YII=B*SIN(VI(I))
      XD=XII-XI(MP,MP,1)
      YD=YII-XI(MP,MP,2)
      S=SQRT(XD*XD+YD*YD)
      ZII=S*CTHS/STHS+XI(MP,MP,3)
C!!!  IS GRAZING INC. POINT OFF OF FINITE CYLINDER?
      IF(ZII.GT.ZC(1)+XII*CTC(1).OR.
     2ZII.LT.ZC(2)+XII*CTC(2)) GO TO 1
      ZD=ZII-XI(MP,MP,3)
      S=SQRT(S*S+ZD*ZD)
      CALL RADCV(RCI,RT,VI(I))
      GM=(PI*RCI)**(1./3.)
      DO 36 NJ=1,3
      DO 36 NI=1,3
36    VAX(NI,NJ)=VXI(NI,NJ,MP)
      CALL SOURCE(EF,EG,EIX,EIY,EIZ,THSR,PHSR,VAX)
      CF=CEXP(CJ*TPI*(XI(MP,MP,1)*D(1)+XI(MP,MP,2)*D(2)+
     2XI(MP,MP,3)*D(3)))
      BX=-UN(2)*D(3)
      BY=UN(1)*D(3)
      BZ=UN(2)*D(1)-UN(1)*D(2)
C!!!  CALCULATE HARD AND SOFT COMPONENTS OF INCIDENT FIELD
      EHP=EIX*UN(1)+EIY*UN(2)
      EST=EIX*BX+EIY*BY+EIZ*BZ
      IF(I.EQ.1) EHP=-EHP
      IF(I.EQ.2) EST=-EST
      CFH=GM*CPI4*QFUN(0.)/SQRT(PI*S)
      CFS=GM*CPI4*PFUN(0.)/SQRT(PI*S)
C!!!  CALCULATE GRAZING INCIDENCE TRANSITION FIELD
      DETH=(0.5*EF*SGN-CFS*EST)*CF
      DEPH=(0.5*EG*SGN-CFH*EHP)*CF
6     CONTINUE
C!!!  CALCULATE TOTAL CYLINDER FIELDS
      EP=EP+DEPH
      ET=ET+DETH
      IF(LDEBUG) WRITE(6,*) I,SXMIG,)X,FI,CF
      IF(LDEBUG) WRITE(6,*) CFH,CFS
      IF(LDEBUG) WRITE(6,*) EHT,EST
      IF(LDEBUG) WRITE(6,*) EHP,ESP
      IF(LDEBUG) WRITE(6,*) DETH,DEPH
1     I=I+1
      IF(I.LE.2) GO TO 3
909   CONTINUE
      IF(.NOT.LTEST) RETURN
      WRITE(6,910)
910   FORMAT(/,' TESTING RPLSCL SUBROUTINE')
      WRITE(6,*) ET,EP,MP
      WRITE(6,*) EHT,EHP
      RETURN
      END
```

SCLRPL

PURPOSE

To compute the far-zone electric field of a source ray which is scattered by the cylinder and then reflected by a given plate.

PERTINENT GEOMETRY

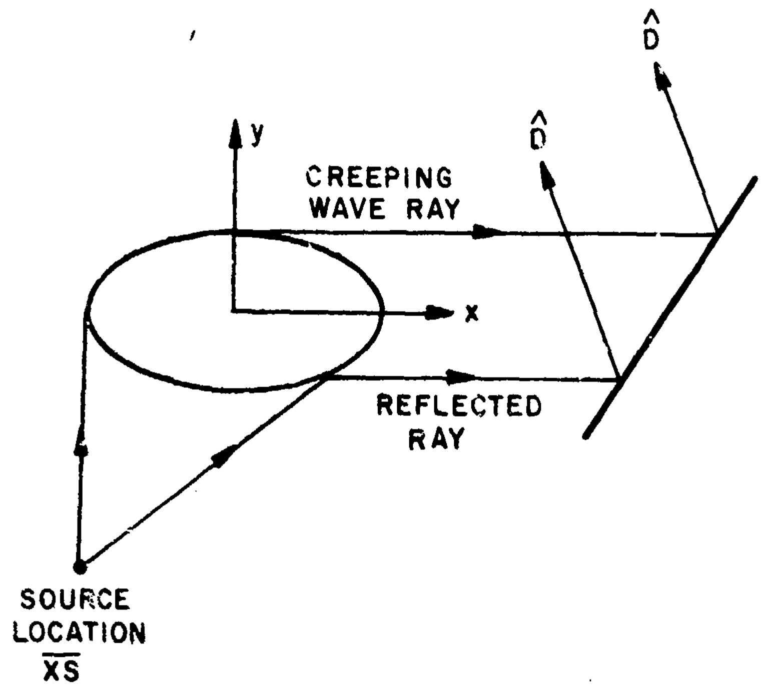


Figure 102--Illustration of ray scattered by the cylinder and reflected by a plate

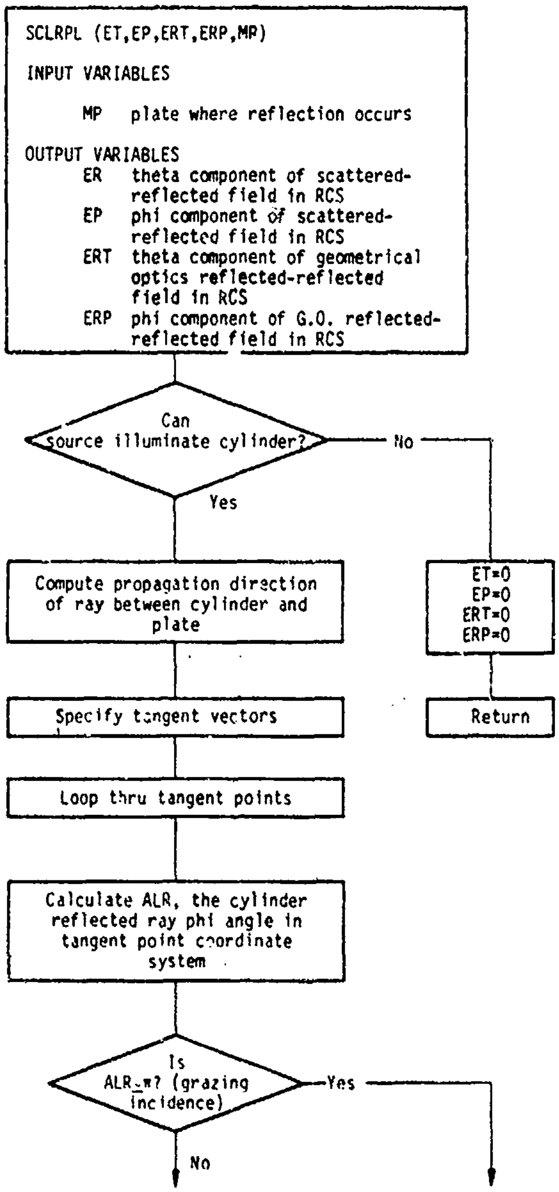
METHOD

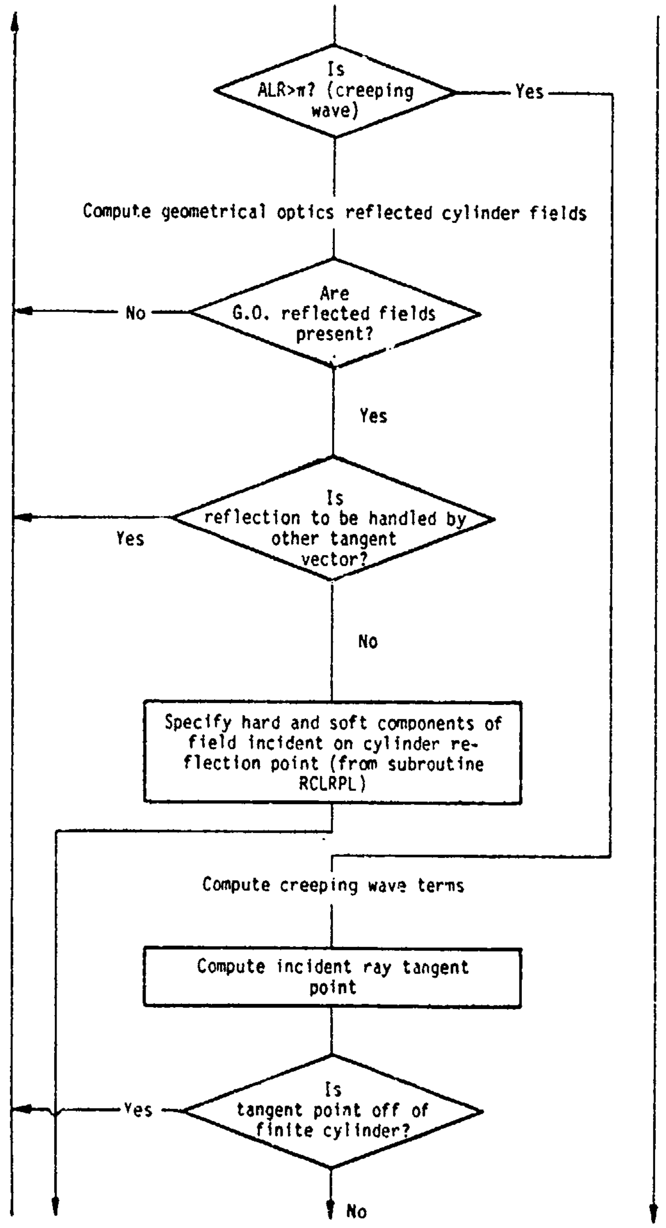
A uniform Geometrical Theory of Diffraction solution for the field reflected or diffracted by a cylinder then reflected by a plate is computed in this subroutine. The fields reflected or diffracted by the cylinder in the direction of the plate are determined in a similar manner as the fields calculated in subroutine SCTCYL. The direction of the ray incident on the plate is determined by imaging the observation direction into the plate, as illustrated in Figure 102. The plate reflected fields are found by satisfying the boundary conditions for the fields at the surface of the plate. The phase of the resultant scattered-reflected fields are referred to the reference coordinate origin. The form of this field is then given by

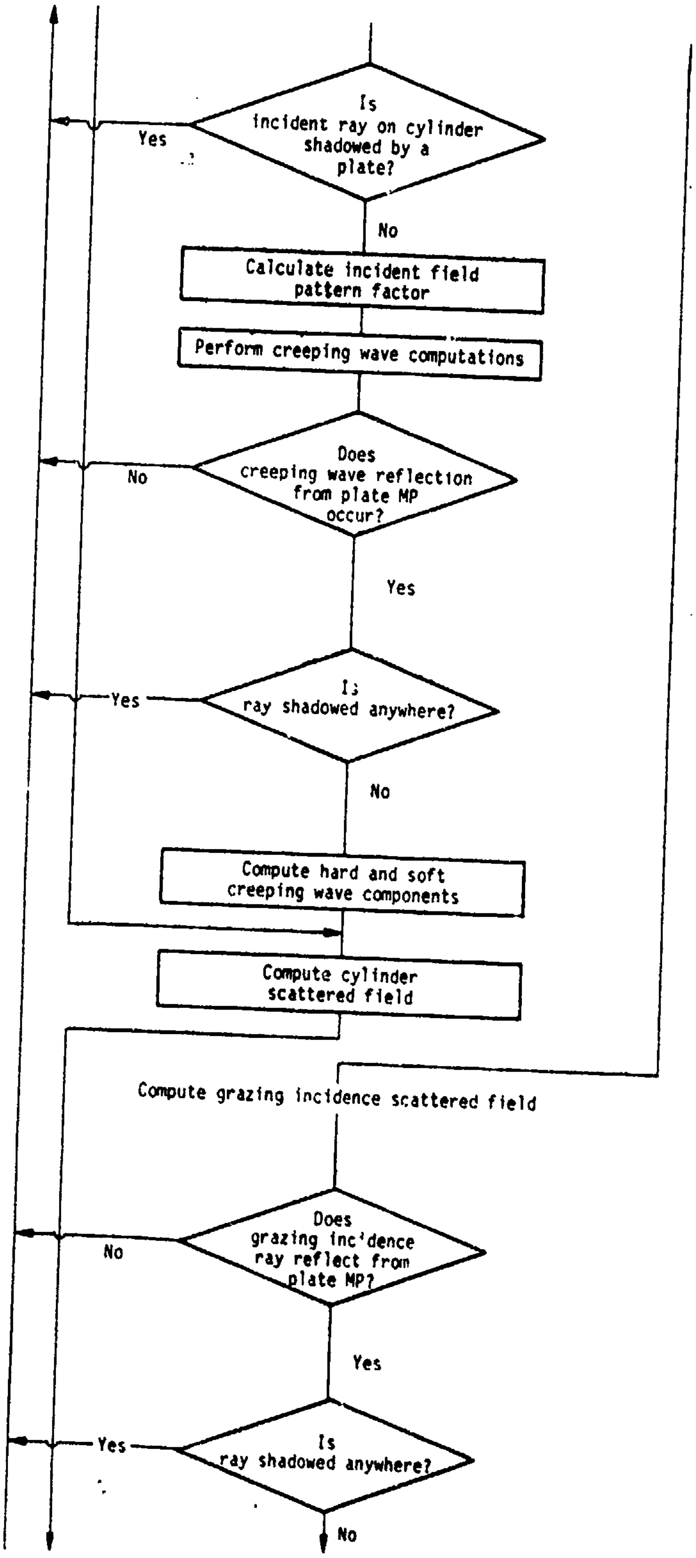
$$\overline{E}^{s,r} = W_m(ET\hat{\theta}+EP\hat{\phi})\ \frac{e^{-jkR}}{R}\quad ,$$

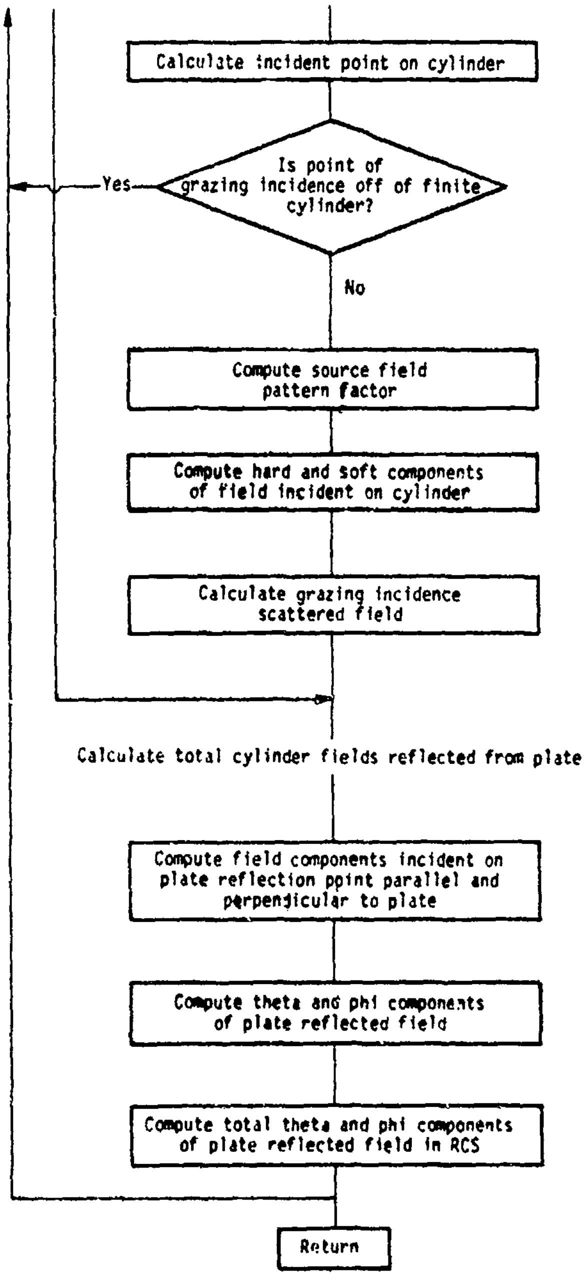
where the factor $\frac{e^{-jkR}}{R}$ and the source weight ($W_m$) are added elsewhere in the code.

FLOW DIAGRAM


SCLRPL (ET,EP,ERT,ERP,MP)
INPUT VARIABLES
MP plate where reflection occurs
OUTPUT VARIABLES
ER theta component of scattered-reflected field in RCS
EP phi component of scattered-reflected field in RCS
ERT theta component of geometrical optics reflected-reflected field in RCS
ERP phi component of G.O. reflected-reflected field in RCS
Can source illuminate cylinder?
No
Yes
Compute propagation direction of ray between cylinder and plate
ET=0
EP=0
ERT=0
ERP=0
Specify tangent vectors
Return
Loop thru tangent points
Calculate ALR, the cylinder reflected ray phi angle in tangent point coordinate system
Is ALR≃π? (grazing incidence)
Yes
No


Is ALR>π? (creeping wave)
Yes
Compute geometrical optics reflected cylinder fields
Are G.O. reflected fields present?
No
Yes
Is reflection to be handled by other tangent vector?
Yes
No
Specify hard and soft components of field incident on cylinder reflection point (from subroutine RCLRPL)
Compute creeping wave terms
Compute incident ray tangent point
Is tangent point off of finite cylinder?
Yes
No


Is
incident ray on cylinder
shadowed by a
plate?
Yes
No
Calculate incident field
pattern factor
Perform creeping wave computations
Does
creeping wave reflection
from plate MP
occur?
No
Yes
Is
ray shadowed anywhere?
Yes
No
Compute hard and soft
creeping wave components
Compute cylinder
scattered field
Compute grazing incidence scattered field
Does
grazing incidence
ray reflect from
plate MP?
No
Yes
Is
ray shadowed anywhere?
Yes
No


Calculate incident point on cylinder
Is point of grazing incidence off of finite cylinder?
Yes
No
Compute source field pattern factor
Compute hard and soft components of field incident on cylinder
Calculate grazing incidence scattered field
Calculate total cylinder fields reflected from plate
Compute field components incident on plate reflection ppint parallel and perpenjicular to plate
Compute theta and phi components of plate reflected field
Compute total theta and phi components of plate reflected field in RCS
Return

## SYMBOL DICTIONARY

| | |
|---|---|
| A1 | FIELD COMPONENTS OF RAY INCIDENT ON PLATE |
| A2 | NORMAL AND TANGENT TO PLATE |
| A3 | DETERMINANT OF POLARIZATION TRANSFORMATION |
| ALR | PHI ANGLE DEFINING PROPAGATION DIRECTION IN TAN |
| | POINT COORDINATE SYSTEM (2-D) |
| ALRS | DIFFERENCE BETWEEN ALS AND ALR |
| ALS | PHI ANGLE DEFINING DIRECTION OF RAY FROM RCS |
| | ORIGIN TO SOURCE IN TAN POINT COORD SYS |
| BX | X,Y,Z COMPONENTS OF POLARIZATION UNIT VECTOR |
| BY | OF SOFT COMPONENT CF FIELD INCIDENT ON CYL (PARALLEL |
| BZ | TO CYL SURFACE AND NORMAL TO INC FIELD PROP DIR) |
| C11 | |
| C12 | COEFFICIENTS USED TO CONVERT POLARIZATION FROM |
| C21 | THETA AND PHI COMPONENTS IN RCS TO COMPONENTS |
| C22 | NORMAL AND TANGENT TO PLATE (AND VICE-VERSA) |
| CFH | HARD TRANSITION FIELD COEFFICIENT |
| CFS | SOFT TRANSITION FIELD COEFFICIENT |
| DEPH | PHI COMPONENT OF TRANSITION FIELD IN RCS |
| DETH | THETA COMPONENT OF TRANSITION FIELD IN RCS |
| DI | X,Y, AND Z COMPONENTS OF INCIDENT RAY DIRECTION ON CYL IN RCS |
| DJ | X,Y,Z COMPONENTS OF PROPAGATION DIRECTION OF RAY |
| | BETWEEN CYLINDER AND PLATE IN RCS |
| EF | THETA COMPONENT OF SOURCE FIELD PATTERN FACTOR IN RCS |
| EG | PHI COMPONENT OF SOURCE FIELD PATTERN FACTOR IN RCS |
| EHP | PHI COMPONENT OF HARD COMPONENT OF GEOMETRICAL |
| | OPTICS FIELD INCIDENT ON CYLINDER IN RCS |
| EHT | THETA COMPONENT OF HARD COMPONENT OF GEOMETRICAL |
| | OPTICS FIELD INCIDENT ON CYLINDER IN RCS |
| ER | DOT PRODUCT OF CYLINDER TANGENT UNIT VECTOR AND REFLECTED |
| | RAY PROPAGATION DIRECTION (2-D) |
| ERP | PHI COMPONENT OF G.O. REFL-REFL FIELD IN RCS |
| ERT | THETA COMPONENT OF G.O. REFL-REFL FIELD IN RCS |
| ESP | PHI COMPONENT OF SOFT COMPONENT OF GEOMETRICAL |
| | OPTICS FIELD INCIDENT ON CYLINDER IN RCS |
| EST | THETA COMPONENT OF SOFT COMPONENT OF GEOMETRICAL |
| | OPTICS FIELD INCIDENT ON CYLINDER IN RCS |
| PHIR | PHI COMPONENT OF INCIDENT RAY DIRECTION ON CYL |
| PHJR | PHI COMPONENT OF RAY PROPAGATION DIRECTION BETWEEN |
| | CYLINDER AND PLATE |
| SKWIG | PARAMETER USED IN TRANSITION FUNCTION |
| THIR | THETA COMPONENT OF INCIDENT RAY DIRECTION ON CYLINDER |
| THJR | THETA COMPONENT OF RAY PROPAGATION DIRECTION BETWEEN CYLINDER |
| | AND PLATE |
| TIRM | PARAMETER USED IN TRANSITION FUNCTION |
| TX1 | X,Y COMPONENTS OF RAY FROM SOURCE |
| TY1 | TANGENT TO TAN POINT 1 (2-D) |
| TX2 | X,Y COMPONENTS OF RAY FROM SOURCE |
| TY2 | TANGENT TO TAN POINT 2 (2-D) |
| UB | X,Y COMPONENTS OF UNIT VECTOR TANGENT TO CYL AT TAN POINT |
| UN | X,Y COMPONENTS OF UNIT VECTOR NORMAL TO CYL AT TAN POINT |
| VI | ELL. ANGLE USED TO DEFINE TANGENT POINTS (2-D) |
| VL | ELL ANGLE DEFINING LOWER LIMIT OF CREEPING WAVE TRAVEL |
| | ON CYLINDER |
| VT | X,Y,Z COMPONENTS OF POLARIZATION UNIT VECTOR PERPENDICULAR |
| | TO PLANE OF INCIDENCE FOR RAY INCIDENT ON PLATE |
| VU | ELL ANGLE DEFINING UPPER LIMIT OF CREEPING WAVEL TRAVEL |
| | ON CYLINDER |
| XD | X,Y,Z COMPONENTS OF DIRECTION OF RAY FROM |
| YD | SOURCE TO CYLINDER TANGENT POINT (INCIDENT |
| ZD | RAY FOR CREEPING AND GRAZING INC. CASES) |
| XI | |
| YI | X,Y,Z COMPONENTS OF POINT WHERE INCIDENT CREEPING |
| ZI | WAVE (OR GRAZING WAVE) MEETS CYLINDER |
| XRF | X,Y,Z COMPONENTS OF POINT WHERE CREEPING WAVE LEAVES CYLINDER |
| XRS | X,Y,Z COMPONENTS OF REFLECTION POINT LOCATION ON PLATE MP |
| | ALSO POINT WHERE CREEPING WAVE LEAVES CYLINDER |
| | ALSO IMAGE OF XRF IN PLATE MP |

CODE LISTING

```
C-----------------------------------------------------------------------
      SUBROUTINE SCLRPL(ET,EP,ERT,ERP,NP)
C!!!
C!!!  COMPUTES THE FIELD SCATTERED FROM THE CYLINDER THEN REFLECTED
C!!!  FROM PLATE #NP
C!!!
      COMPLEX CJ,CPI4,CF,CFH,CFS,FI,PFUN,CFUN
      COMPLEX EIX,EIY,EIZ,EIPH,EITH,ET,EP,ERT,ERP
      COMPLEX REF,ESTH,ESPH,EHTH,EHPH,DETH,DEPH,EF,EG
      COMPLEX EST,ESP,EHT,EHP,A1,A2
      DIMENSION VI(2),ER(2),UN(2),UB(2),DI(3),XRF(3),XRS(3),VT(3),DJ(3)
      LOGICAL LHIT,LTRFJ,LDEBUG,LTEST,LRFS,LRFST
      COMMON/GEOPLA/X(14,6,3),V(14,6,3),VP(14,6,3),VN(14,3)
     2,MEP(14),MPX
      COMMON/GEOMEL/A,B,ZC(2),SNC(2),CNC(2),CTC(2)
      COMMON/SORINF/XS(3),VXS(3,3)
      COMMON/FIS/PI,TPI,DPR,RPD
      COMMON/GTD/AS,ID,SAS,SASP,CAS
      COMMON/DIR/D(3),THSR,PHSR,SPS,CPS,STHS,CTHS
      COMMON/THPHUV/DT(3),DP(2)
      COMMON/COMP/CJ,CPI4
      COMMON/LNDSCL/DTS,VTS(2),BTS(4)
      COMMON/FUDGJ/REF,ESTH,ESPH,EHTH,EHPH,XR(3),RG,RHO1,DL,LTRFJ
      COMMON/TEST/LDEBUG,LTEST
      COMMON/CLRFS/LRFS(14)
      EXTERNAL FCT
      ET=(0.,0.)
      EP=(0.,0.)
      ERT=(0.,0.)
      ERP=(0.,0.)
C!!!  CAN SOURCE ILLUMINATE CYLINDER?
      IF(DTS.LT.-1.5) GO TO 999
C!!!  COMPUTE PROPAGATION DIRECTION OF RAY BETWEEN
C!!!  CYLINDER AND PLATE
      CALL REFIP(PHJR,THJR,PHSR,THSR,NP)
      SPHJ=SIN(PHJR)
      CPHJ=COS(PHJR)
      STHJ=SIN(THJR)
      CTHJ=COS(THJR)
      DJ(1)=CPHJ*STHJ
      DJ(2)=SPHJ*STHJ
      DJ(3)=CTHJ
      AS=PI-THJR
      SAS=SIN(AS)
      SASP=ABS(SIN(AS-0.5*PI))
      CAS=COS(AS)
C!!!  SPECIFY TANGENT VECTORS
      TX1=BTS(1)
      TY1=BTS(2)
      TX2=BTS(3)
      TY2=BTS(4)
      ER(1)=TX1*CPHJ+TY1*SPHJ
      ER(2)=TX2*CPHJ+TY2*SPHJ
C!!!  LOOP THRU TANGENT VECTORS
      I=1
      LRFST=.FALSE.
      VI(1)=VTS(1)
      VI(2)=VTS(2)
      IF(LDEBUG) WRITE(6,900)
  900 FORMAT(/,' DEBUGGING SCLRPL SUBROUTINE')
    1 CONTINUE
      CALL NAFUB(UN,UB,VI(I))
      SINA=UN(1)*CPHJ+UN(2)*SPHJ
C!!!  CALCULATE ALR, THE REFL RAY PHI ANGLE IN TAN POINT COORD SYS.
      ALR=ATAN2(SINA,-ER(I))
      IF(ALR.LT.0.) ALR=ALR+TPI
```

```
C!!!  IF GRAZING INCIDENCE IS PRESENT, SKIP TO APPPROPRIATE SECTION
      IF(ABS(PI-ALR).LT.0.0085) GO TO 5
C!!!  IF ALR IS G.T. THAN PI, COMPUTE CREEPING WAVE TERMS
      IF(ALR.GT.PI) GO TO 10
C!!!  COMPUTE G.O. REFLECTED FIELD TERMS IF ALR .LE. PI
      CALL RCLRPL(ERT,ERP,MP)
C!!!  ARE REFLECTED FIELDS PRESENT?
      IF(LIRFJ) GO TO 1
      SNAS=UN(1)*XS(1)+UN(2)*XS(2)
      IC=2*I-1
      CSAS=BTS(IC)*XS(1)+BTS(IC+1)*XS(2)
      ALS=BTAN2(SNAS,-CSAS)
      ALRS=ALR-ALS
C!!!  IS REFLECTION TO BE HANDLED WITH OTHER TAN VECTOR?
      IF(ABS(ALRS).LT.0.0085.AND.I.EQ.2) GO TO 1
      IF(ALRS.LE.-0.0085) GO TO 1
      GM=(PI*RC)**(1./3.)
      RHS=RHO1*DL/(DL-RHO1)
      SKWIG=-ABS(2.*TPI*RHS/GM/GM)
      CF=-SQRT(-2./PI/SKWIG)*CPI4*REF
      CF=CF*CEXP(-CJ*SKWIG*SKWIG*SKWIG/12.)
      TTRM=SKWIG/GM
      XX=PI*(DL+RHS)*TTRM*TTRM
C!!!  SPECIFY G.O. REFLECTED FIELD COMPONENTS (FROM REFCYL)
      EST=ESTR
      ESP=ESPR
      EHT=EHTR
      EHP=EHPR
      GO TO 30
   10 CONTINUE
      IF(LRFST) LRFS(MP)=.FALSE.
      LRFST=.TRUE.
C!!!  COMPUTE CREEPING WAVE TERMS IF ALR .GT. PI
C!!!  COMPUTE INCIDENT RAY TANGENT POINT
      XI=A*COS(VI(I))
      YI=B*SIN(VI(I))
      XD=XI-XS(1)
      YD=YI-XS(2)
      S=SQRT(XD*XD+YD*YD)
      ZI=S*CTHJ/STHJ+XS(3)
C!!!  IS TANGENT POINT ON CYLINDER?
      IF(ZI.GT.ZC(1)+XI*CTC(1).OR.
     2ZI.LT.ZC(2)+YI*CTC(2)) GO TO 1
      ZD=ZI-XS(3)
      PHIR=BTAN2(YD,XD)
      THIR=BTAN2(S,ZD)
      S=SQRT(S*S+ZD*ZD)
      DI(1)=XD/S
      DI(2)=YD/S
      DI(3)=ZD/S
C!!!  DOES INCIDENT RAY HIT PLATE BEFORE CYLINDER?
      CALL PLAINT(XS,DI,DHIT,P,LHIT)
      IF(LHIT.AND.(DHIT.LT.S)) GO TO 1
C!!!  CALCULATE INCIDENT FIELD PATTERN FACTOR
      CALL SOURCE(EF,EG,EIX,EIY,EIZ,THIR,PHIR,VXS)
      IF(LDEBUG) WRITE(6,*) EF,EG
C!!!  PERFORM CREEPING WAVE COMPUTATIONS
      IF(I.EQ.1) VD=BTAN2(-B*CPHJ,A*SPHJ)
      IF(I.EQ.2) VD=BTAN2(B*CPHJ,-A*SPHJ)
      VDP=VD-VI(I)
      IF(VDP.GT.PI) VDP=VDP-TPI
      IF(VDP.LT.-PI) VDP=VDP+TPI
      IF(I.EQ.2) GO TO 20
      IF(VDP.LT.0.) GO TO 1
      VL=VI(I)
      VU=VDP+VI(I)
```

```
         GO TO 25
    20   CONTINUE
         IF(VDP.GT.0.) GO TO 1
         VL=VDP+VI(I)
         VU=VI(I)
    25   CONTINUE
         CALL FXARG(SKWIG,AS,VL,VU)
         XHF(1)=A*COS(VD)
         XHF(2)=B*SIN(VD)
         ID=3
         CALL DQG32(VL,VU,FCT,SS)
         SS=SS/SAS
         XHF(3)=ZI+SS*CTHJ
         DO 26 N=1,3
    26   XRS(N)=XHF(N)
  C!!!   DOES CREEPING WAVE REFLECTION FROM PLATE
  C!!!   MP OCCUR?
         CALL PLAINT(XRS,DJ,DHJT,-MP,LHIT)
         IF(.NOT.LHIT) GO TO 1
  C!!!   IS RAY SHADOWED ANYWHERE?
         CALL PLAINT(XRS,D,DHT,MP,LHIT)
         IF(LHIT) GO TO 1
         CALL CYLINT(XRS,D,PHSR,DHT,LHIT,.TRUE.)
         IF(LHIT) GO TO 1
         CALL PLAINT(XHF,DJ,DHT,MP,LHIT)
         IF(LHIT.AND.(DHT.LT.DHJT)) GO TO 1
         CALL RADCV(RGI,RT,VI(I))
         CALL RADCV(RGF,RT,VD)
         GAM=(PI*PI*RGI*RGF)**(1./6.)
         CF=-GAM*CPI4*CEXP(-CJ*TPI*(S+SS))/PI/SQRT(2.*S)
  C!!!   COMPUTE PHASE TERM
         CALL IMAGE(XRS,XHF,AMR,MP)
         CF=CF*CEXP(CJ*TPI*(XRS(1)*D(1)+XRS(2)*D(2)+XRS(3)*D(3)))
         TTRM=SKWIG/GAM
         XX=PI*S*TTRM*TTRM
         BX=-UN(2)*DI(3)
         BY=UN(1)*DI(3)
         BZ=UN(2)*DI(1)-UN(1)*DI(2)
         ESP=(0.,0.)
         EHT=(0.,0.)
  C!!!   COMPUTE HARD AND SOFT CREEPING WAVE COMPONENTS
         EHP=EIX*UN(1)+EIY*UN(2)
         EST=EIX*BX+EIY*BY+EIZ*BZ
         IF(I.EQ.1) EHP=-EHP
         IF(I.EQ.2) EST=-EST
    2    CONTINUE
  C!!!   COMPUTE THE CYLINDER SCATTERED FIELD
         XXS=SQRT(TPI*XX)
         XXX=SQRT(2.*XX/PI)
         CALL FRNELS(CCC,SSS,XXX)
         FI=CMPLX(0.5-CCC,SSS-0.5)
         FI=XXS*FI*CEXP(CJ*(.5*PI+XX))
         FI=-FI/SKWIG/SQRT(2.)
         SQTP=SQRT(2.*PI)
         CPH=CF*(FI+SQTP*CFUN(SKWIG))
         CPS=CF*(FI+SQTP*PFUN(SKWIG))
         CEPH=CPH*EHP+CPS*ESP
         CEST=CPH*EHT+CPS*EST
         GO TO 6
    5    CONTINUE
  C!!!   COMPUTE GRAZING INCIDENCE SCATTERED FIELD
         DO 7 N=1,3
    7    XRS(N)=XS(N)
  C!!!   DOES REFLECTION FROM PLATE MP OCCUR?
         CALL PLAINT(XRS,DJ,DHJT,-MP,LHIT)
         IF(.NOT.LHIT) GO TO 1
```

```
C!!!  IS RAY SHADOWED ANYWHERE?
      CALL PLAINT(XRS,D,DHT,NP,LHIT)
      IF(LHIT) GO TO 1
      CALL CYLINT(XRS,D,PHSR,DHT,LHIT,.TRUE.)
      IF(LHIT) GO TO 1
      CALL PLAINT(XS,DJ,DHT,NP,LHIT)
      IF(LHIT.AND.(DHT.LT.DHJT)) GO TO 1
      SGN=-SIGN(1.,SIN(ALR))
C!!!  CALCULATE INCIDENT POINT
      XI=A*COS(VI(I))
      YI=B*SIN(VI(I))
      XD=XI-XS(1)
      YD=YI-XS(2)
      S=SQRT(XD*XD+YD*YD)
      ZI=S*CTHJ/STHJ+XS(3)
C!!!  IS POINT OF GRAZING INCIDENCE OFF OF FINITE CYLINDER?
      IF(ZI.GT.ZC(1)+XI*CTC(1).OR.
     2ZI.LT.ZC(2)+XI*CTC(2)) GO TO 1
      ZD=ZI-XS(3)
      S=SQRT(S*S+ZD*ZD)
      CALL RADCV(RGI,RT,VI(I))
      GM=(PI*RGI)**(1./3.)
C!!!  CALCULATE INCIDENT FIELD PATTERN FACTOR
      CALL SOURCE(EF,EG,EIX,EIY,EIZ,THIR,PHIR,VXS)
C!!!  CALCULATE PHASE TERM
      CALL IMAGE(XRS,XS,ANR,NP)
      CF=CEXP(CJ*TPI*(XRS(1)*D(1)+XRS(2)*D(2)+XRS(3)*D(3)))
      BX=-UN(2)*DJ(3)
      BY=UN(1)*DJ(3)
      BZ=UN(2)*DJ(1)-UN(1)*DJ(2)
C!!!  CALCULATE HARD AND SOFT COMPONENTS OF CYL FIELD
      EHP=EIX*UN(1)+EIY*UN(2)
      EST=EIX*BX+EIY*BY+EIZ*BZ
      IF(I.EQ.1) EHP=-EHP
      IF(I.EQ.2) EST=-EST
      CFH=GM*CPI4*QFUN(0.)/SQRT(PI*S)
      CFS=GM*CPI4*PFUN(0.)/SQRT(PI*S)
C!!!  CALCULATE GRAZING INCIDENCE SCATTERED FIELD
      DETH=(0.5*EF*SGN-CFS*EST)*CF
      DEPH=(0.5*EG*SGN-CFH*EHP)*CF
    6 CONTINUE
C!!!  CALCULATE TOTAL CYLINDER FIELDS REFLECTED FROM PLATE
      VT(1)=VN(NP,2)*D(3)-VN(NP,3)*D(2)
      VT(2)=VN(NP,3)*D(1)-VN(NP,1)*D(3)
      VT(3)=VN(NP,1)*D(2)-VN(NP,2)*D(1)
      C11=VN(NP,1)*CPHJ*CTHJ+VN(NP,2)*SPHJ*CTHJ-VN(NP,3)*STHJ
      C12=-VN(NP,1)*SPHJ+VN(NP,2)*CPHJ
      C21=VT(1)*CPHJ*CTHJ+VT(2)*SPHJ*CTHJ-VT(3)*STHJ
      C22=-VT(1)*SPHJ+VT(2)*CPHJ
      A1=DETH*C11+DEPH*C12
      A2=DETH*C21+DEPH*C22
      C11=VN(NP,1)*DT(1)+VN(NP,2)*DT(2)+VN(NP,3)*DT(3)
      C12=VN(NP,1)*DP(1)+VN(NP,2)*DP(2)
      C21=VT(1)*DT(1)+VT(2)*DT(2)+VT(3)*DT(3)
      C22=VT(1)*DP(1)+VT(2)*DP(2)
      A3=C11*C22-C12*C21
      DETH=(A1*C22+A2*C12)/A3
      DEPH=-(A2*C11+A1*C21)/A3
      EP=EP+DEPH
      ET=ET+DETH
      IF(LDEBUG) WRITE(6,*) I,SKIC,JX,FI,CF
      IF(LDEBUG) WRITE(6,*) CFH,CFS
      IF(LDEBUG) WRITE(6,*) EHT,EST
      IF(LDEBUG) WRITE(6,*) EHP,ESP
      IF(LDEBUG) WRITE(6,*) DETH,DEPH
    1 I=I+1
```

```
      IF(I.LE.2) GO TO 3
509   CONTINUE
      IF(.NOT.LTEST) RETURN
      WRITE(6,510)
510   FORMAT(/,' TESTING SCLRPL SUBROUTINE')
      WRITE(6,*) ET,EP,MP
      WRITE(6,*) ERT,ERP
      RETURN
      END
```

## SCTCYL

### PURPOSE

To calculate the far-zone fields scattered by the elliptic cylinder's curved surface.

### PERTINENT GEOMETRY

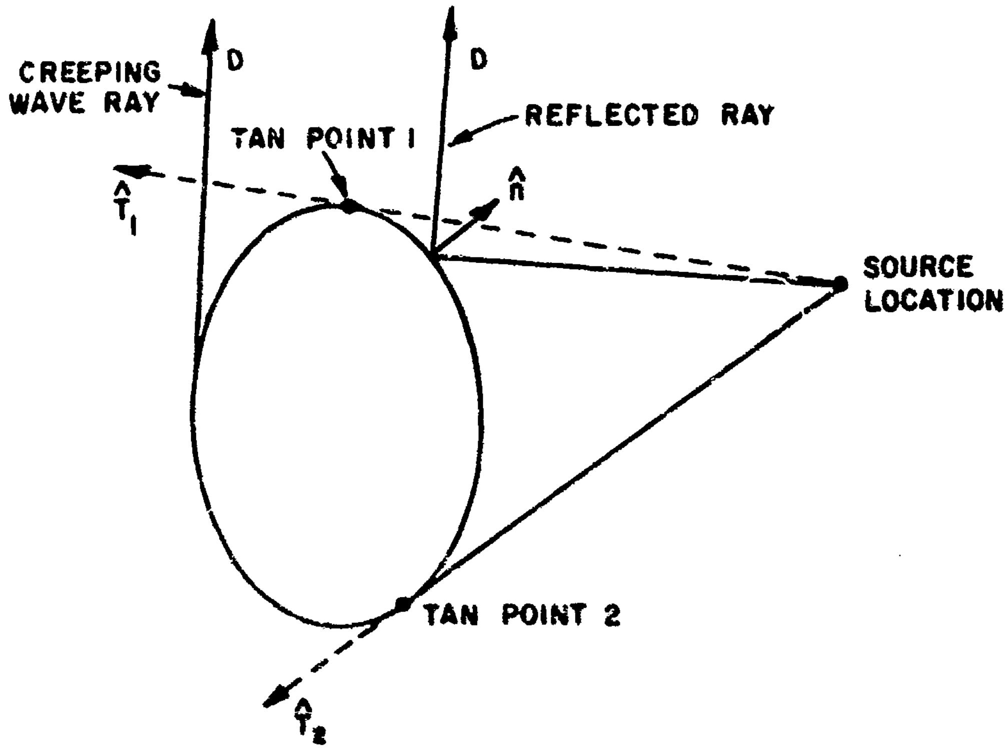


Figure 103--Illustration of reflected and creeping wave scattering by the elliptic cylinder.

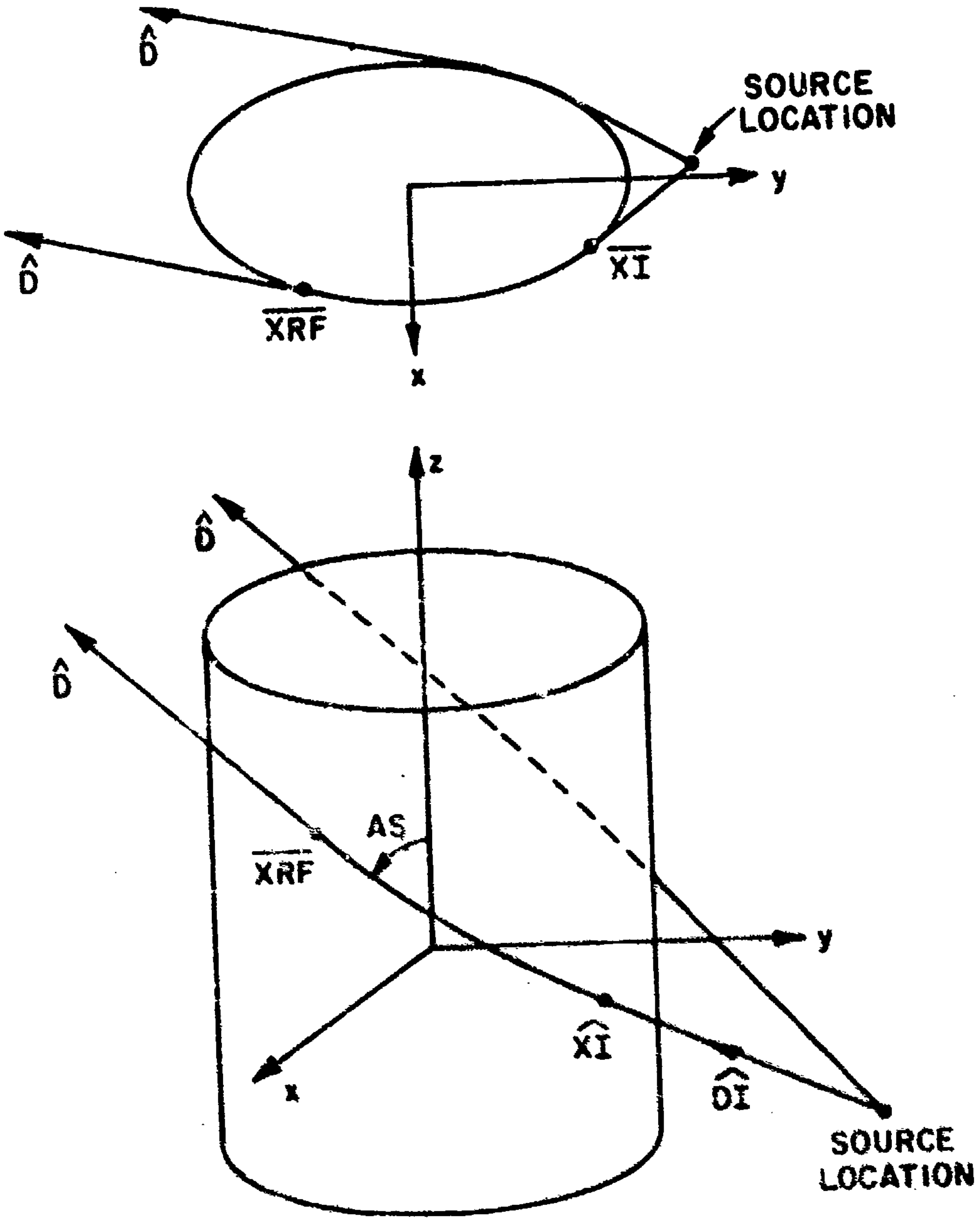


Figure 104--Geometry of creeping wave scattering.

$$\overline{XRF} = \hat{x}\ XRF(1) + \hat{y}\ XRF(2) + \hat{z}\ XRF(3)$$

$$\overline{XI} = \hat{x}\ XI(1) + \hat{y}\ XI(2) + \hat{z}\ XI(3)$$

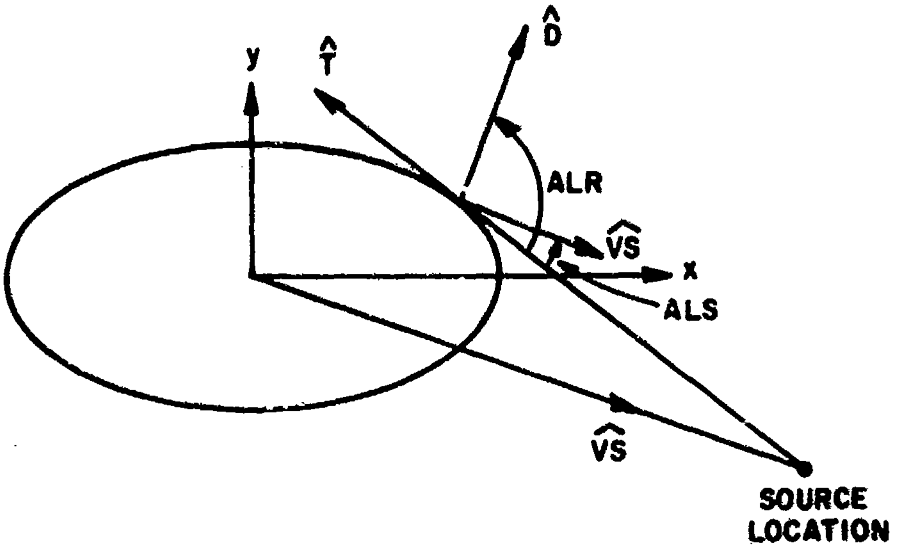


Figure 105--Geometry of angles of cylinder scattering problem.

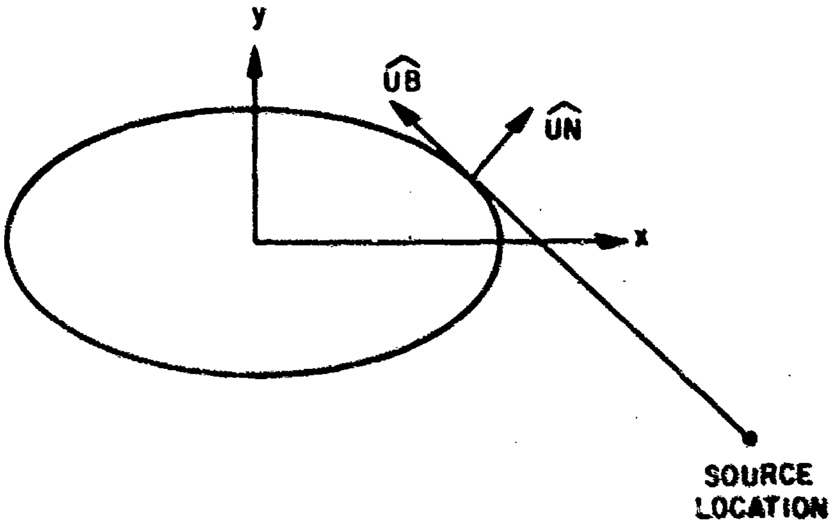


Figure 106--Illustration of tan point coordinate system.

$\widehat{UN} = \hat{x}\ UN(1) + \hat{y}\ UN(2)$

$\widehat{UB} = \hat{x}\ UB(1) + \hat{y}\ UB(2)$

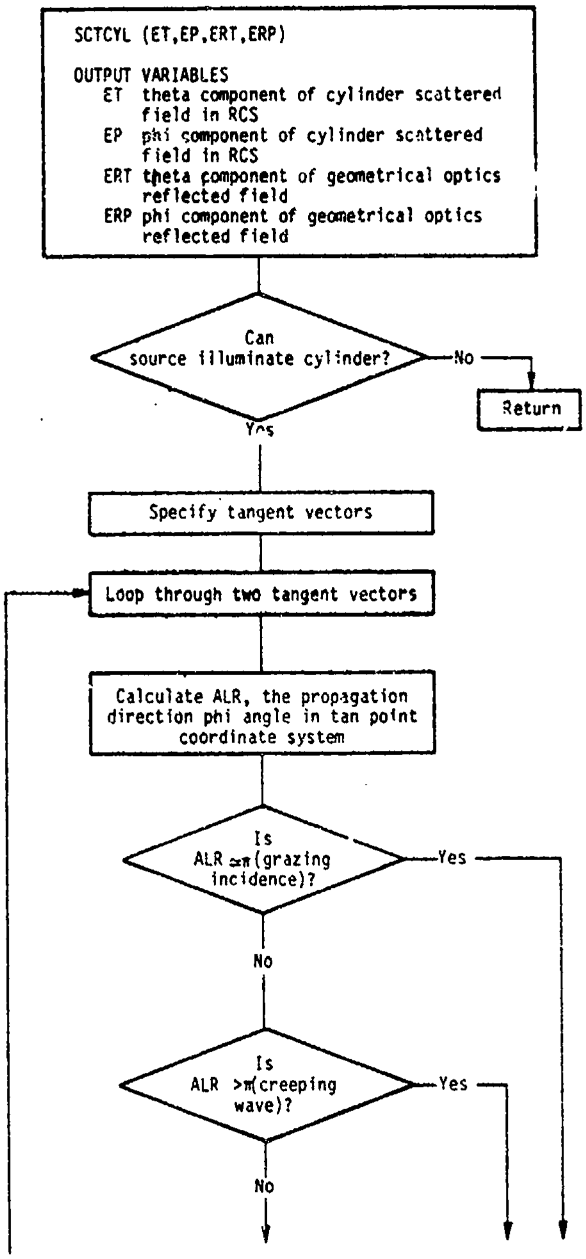
## METHOD

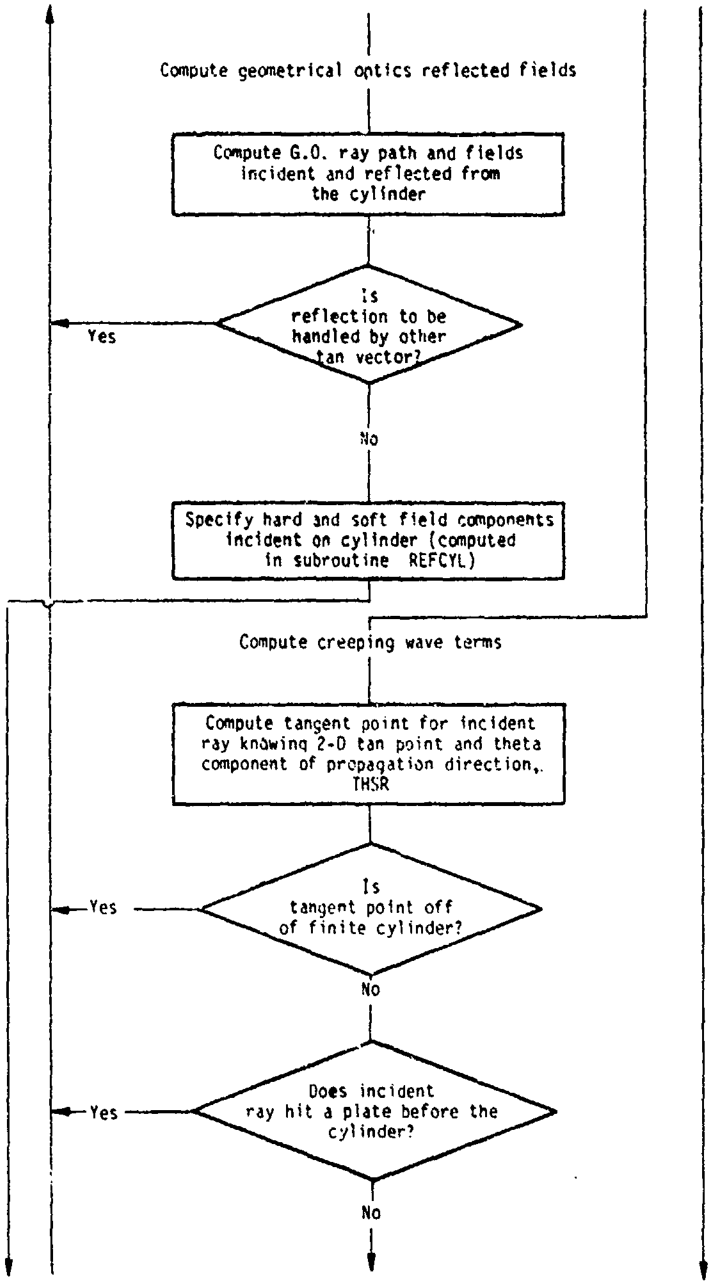
A uniform Geometrical Theory of Diffraction solution[6] is used to compute the reflected and diffracted fields of a source in the presence of the curved surface of an elliptic cylinder. In a given observation direction the solution contains two terms. In the lit region the solution is composed of a reflected field and the dominant creeping wave field, as illustrated in Figure 103. In the shadow region the solution is composed of a clockwise and a counterclockwise creeping wave field, as illustrated in Figure 104. The reflected field and creeping wave fields are modified versions of the usual GTD solution, that is, they are obtained from a uniform solution that is valid at the shadow boundaries (tangent point vector regions) and that goes to the geometrical optics solution in the deep lit region and the usual creeping wave solution in the deep shadow region. The solution is presented in Reference 6 and on pages 112-113 of Reference 1. The phases of the reflected and creeping wave (or transition) fields are referred to the reference coordinate system origin. The fields are combined and the total field scattered by the cylinder is given by

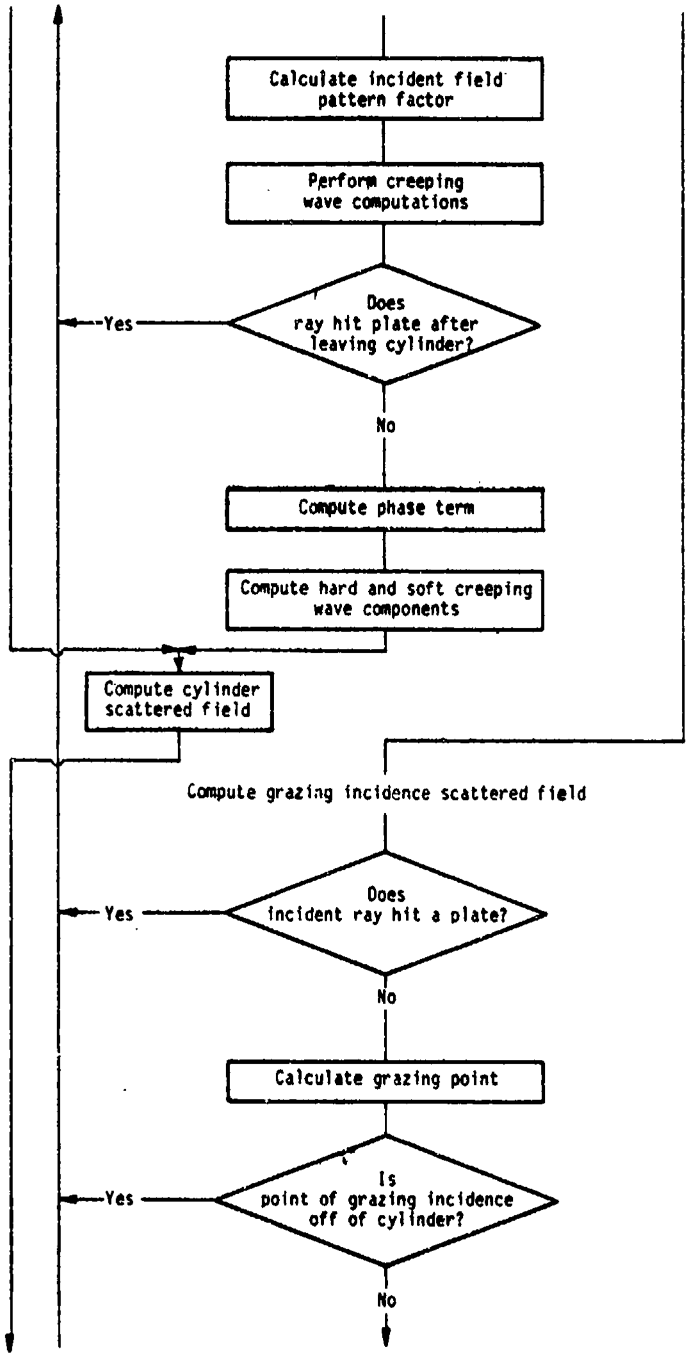
$$\overline{E}^S = W_m(ET\hat{\theta} + EP\hat{\phi}) \frac{e^{-jkR}}{R} ,$$

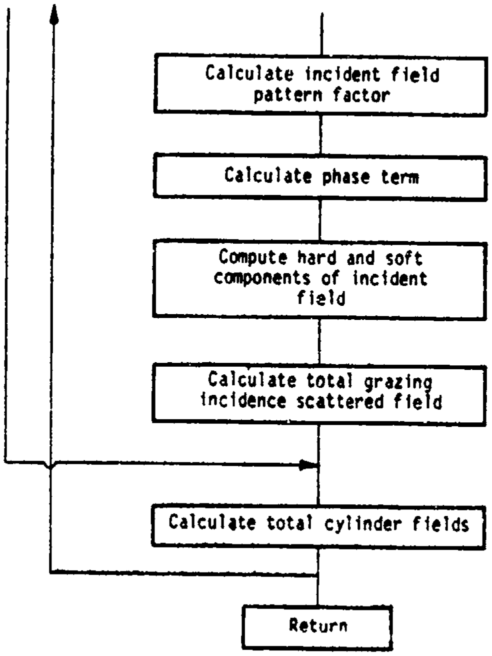
where the factor $\frac{e^{-jkR}}{R}$ and the source weight ($W_m$) are added elsewhere in the code.

## FLOW DIAGRAM


SCTCYL (ET,EP,ERT,ERP)
OUTPUT VARIABLES
ET theta component of cylinder scattered field in RCS
EP phi component of cylinder scattered field in RCS
ERT theta component of geometrical optics reflected field
ERP phi component of geometrical optics reflected field
Can source illuminate cylinder?
No
Return
Yes
Specify tangent vectors
Loop through two tangent vectors
Calculate ALR, the propagation direction phi angle in tan point coordinate system
Is ALR ≅π (grazing incidence)?
Yes
No
Is ALR >π(creeping wave)?
Yes
No


Compute geometrical optics reflected fields
Compute G.O. ray path and fields incident and reflected from the cylinder
Is reflection to be handled by other tan vector?
Yes
No
Specify hard and soft field components incident on cylinder (computed in subroutine REFCYL)
Compute creeping wave terms
Compute tangent point for incident ray knowing 2-D tan point and theta component of propagation direction, THSR
Is tangent point off of finite cylinder?
Yes
No
Does incident ray hit a plate before the cylinder?
Yes
No


Calculate incident field pattern factor
Perform creeping wave computations
Does ray hit plate after leaving cylinder?
Yes
No
Compute phase term
Compute hard and soft creeping wave components
Compute cylinder scattered field
Compute grazing incidence scattered field
Does incident ray hit a plate?
Yes
No
Calculate grazing point
Is point of grazing incidence off of cylinder?
Yes
No


Calculate incident field pattern factor
Calculate phase term
Compute hard and soft components of incident field
Calculate total grazing incidence scattered field
Calculate total cylinder fields
Return

## SYMBOL DICTIONARY

| | |
|---|---|
| ALR | PHI ANGLE DEFINING RADIATION DIRECTION IN TAN POINT COORDINATE SYSTEM (2-D) |
| ALRS | DIFFERENCE BETWEEN ALS AND ALR |
| ALS | PHI ANGLE DEFINING DIRECTION OF RAY FROM RCS ORIGIN TO SOURCE IN TANGENT POINT COORD. SYS |
| AS | ANGLE BETWEEN CREEPING WAVE PATH ON CYL AND LINE PARALLEL TO Z AXIS |
| BX<br>BY<br>BZ | X,Y,Z COMPONENTS OF POLARIZATION UNIT VECTOR OF SOFT COMPONENT OF FIELD INCIDENT ON CYL (PARALLEL TO CYL SURFACE AND NORMAL TO INC RAY PROP DIR) |
| CF | COMPLEX PHASE AND RAY SPREADING COEFFICIENT |
| CFH | HARD TRANSITION FIELD COEFFICIENT |
| CFS | SOFT TRANSITION FIELD COEFFICIENT |
| CSAS | DOT PRODUCT OF CYLINDER TANGENT UNIT VECTOR AND VECTOR FROM ORIGIN TO SOURCE |
| D | PROPAGATION DIRECTION UNIT VECTOR FOR RAY SCATTERED FROM CYL IN (X,Y,Z) RCS COMPONENTS |
| DEPH | PHI COMPONENT OF TRANSITION FIELD IN RCS |
| DETH | THETA COMPONENT OF TRANSITION FIELD IN RCS |
| DHIT | DISTANCE FROM SOURCE TO HIT POINT (FROM PLAINT) |
| DI | X,Y,Z COMPONENTS OF UNIT VECTOR OF PROPAGATION DIRECTION OF RAY INCIDENT ON CYLINDER |
| EF | PATTERN FACTOR FOR THETA COMPONENT OF INCIDENT FIELD IN RCS |
| EG | PATTERN FACTOR FOR PHI COMPONENT OF INCIDENT FIELD PATTERN FACTOR IN RCS |
| EHP | PHI COMPONENT OF HARD COMPONENT OF FIELD INCIDENT ON CYL OR CREEPING WAVE FIELD IN RCS |
| EHT | THETA COMPONENT OF HARD COMPONENT OF FIELD INCIDENT ON CYL OR CREEPING WAVE FIELD IN RCS |
| EIX<br>EIY<br>EIZ | X,Y,Z COMPONENTS OF INCIDENT FIELD PATTERN FACTOR |
| EP | PHI COMPONENT OF CYLINDER E FIELD WITH PHASE REFERRED TO RCS ORIGIN |
| ER | DOT PRODUCT OF UNIT VECTOR TANGENT TO CYLINDER AND THE PROPAGATION DIR. UNIT VECTOR |
| ERP | PHI COMPONENT OF G.O. REFLECTED FIELD |
| ERT | THETA COMPONENT OF G.O. REFLECTED FIELD |
| ESP | PHI COMPONENT OF SOFT COMPONENT OF FIELD INCIDENT ON CYL OR CREEPING WAVE FIELD IN RCS |
| EST | THETA COMPONENT OF SOFT COMPONENT OF FIELD INCIDENT ON CYL OR CREEPING WAVE FIELD IN RCS |
| ET | THETA COMPONENT OF CYLINDER E FIELD WITH PHASE REFERRED TO RCS ORIGIN |
| FI | PARAMETER USED IN TRANSITION FUNCTION |
| GM | VARIABLE USED IN TRANSITION FUNCTION |
| I | VARIABLE USED TO STEP THROUGH TANGENT POINTS |
| IC | INDEX VARIABLE |
| LHIT | SET TRUE IF RAY HITS A PLATE (FROM PLAINT) |
| LTRF | (RETURNED FROM RPLRCL) SET TRUE IF G.O. CYLINDER REFLECTED FIELD DOES NOT EXIST |
| PHIR | PHI COMPONENT OF PROPAGATION DIRECTION OF RAY INCIDENT ON CYLINDER |
| ROF | RADIUS OF CURV OF CYL AT POINT XRF IN X-Y PLANE |
| ROI | RADIUS OF CURV OF CYL AT INC RAY POINT ON CYL IN XY PLANE |
| S | LENGTH OF VECTOR FROM SOURCE TO TAN POINT (2 OR 3-D) |
| SINA | DOT PRODUCT OF CYL UNIT NORMAL AND CYL SCATTERED RAY PROPAGATION DIRECTION UNIT VECTOR |
| SKWIG | PARAMETER USED IN TRANSITION FUNCTION |
| SNAS | DOT PRODUCT OF CYL UNIT NORMAL AND VECTOR FROM ORIGIN TO SOURCE |
| THIR | THETA COMPONENT OF PROPAGATION DIRECTION OF RAY INCIDENT ON CYLINDER |

| | |
|---|---|
| TTRM | PARAMETER USED IN TRANSITION FUNCTION |
| TX1<br>TY1 | X AND Y COMPONENTS OF UNIT VECTOR OF RAY FROM SOURCE TANGENT TO TAN POINT 1 OF ELL CYL (2-D) |
| TX2<br>TY2 | X AND Y COMPONENTS OF UNIT VECTOR OF RAY FORM SOURCE TANGENT TO TAN POINT 2 OF ELL CYL (2-D) |
| UB | X,Y COMPONENTS OF UNIT VECTOR TAN TO CYL AT TAN POINT (2-D) |
| UN | X,Y COMPONENTS OF UNIT NORMAL TO CYL AT TAN POINT (2-D) |
| VD | COMPUTATIONAL VARIABLE |
| VDP | COMPUTATIONAL VARIABLE |
| VI | ELL. ANGLE USED TO DEFINE TANGENT POINTS (2-D) |
| VL | ELL ANGLE DEFINING POINT WHERE.CREEPING WAVE MEETS CYLINDER |
| VU | ELL ANGLE DEFINING POINT WHERE CREEPING WAVE LEAVES CYLINDER |
| XD<br>YD<br>ZD | X,Y,Z COMPONENTS OF DIRECTION OF RAY FROM SOURCE TO CYLINDER TANGENT POINT (INCIDENT RAY FOR CREEPING AND GRAZING INC. CASES) |
| XI<br>YI<br>ZI | X,Y,Z COMPONENTS OF POINT WHERE INCIDENT CREEPING WAVE (OR GRAZING WAVE) MEETS CYLINDER |
| XPP | X,Y,Z COMPONENTS OF POINT WHERE RAY LEAVES CYLINDER |
| XRF | X,Y,Z COMPONENTS OF POINT WHERE CREEPING WAVE LEAVES CYLINDER |
| XX | PARAMETER USED IN TRANSITION FUNCTION |
| XXS | PARAMETER USED IN TRANSITION FUNCTION |
| XXX | PARAMETER USED IN TRANSITION FUNCTION |

# CODE LISTING

```
C-------------------------------------------------------------------
      SUBROUTINE SCTCYL(ET,EP,ERT,ERP)
C!!!
C!!!  GTD SCATTERED FIELD OF AN ELLIPTIC CYLINDER
C!!!
      COMPLEX CJ,CPI4,CF,CFH,CFS,FI,PFUN,QFUN
      COMPLEX EIX,EIY,EIZ,EIPH,EITH,ET,EP,ERT,ERP
      COMPLEX REF,ESTH,ESPH,EHTH,EHPH,DETH,DEPH,EF,EG
      COMPLEX EST,ESP,EHT,EHP
      DIMENSION VI(2),ER(2),UN(2),UB(2),DI(3),XRF(3)
      LOGICAL LHIT,LTRF,LDEBUG,LTEST,LRFC,LRFCT
      COMMON/GEOMEL/A,B,ZC(2),SNC(2),CNC(2),CTC(2)
      COMMON/SORINF/XS(3),VXS(3,3)
      COMMON/PIS/PI,TPI,DPR,RPD
      COMMON/GTD/AS,ID,SAS,SASP,CAS
      COMMON/DIR/D(3),THSR,PHSR,SPS,CPS,STHS,CTHS
      COMMON/COMP/CJ,CPI4
      COMMON/ENDSCL/DTS,VTS(2),BTS(4)
      COMMON/FUDG/REF,ESTH,ESPH,EHTH,EHPH,XR(3),RG,RHO1,DL,LTRF
      COMMON/TEST/LDEBUG,LTEST
      COMMON/CLRFC/LRFC
      EXTERNAL FCT
      ET=(0.,0.)
      EP=(0.,0.)
      ERT=(0.,0.)
      ERP=(0.,0.)
C!!!  CAN SOURCE ILLUMINATE CYLINDER SURFACE?
      IF(DTS.LT.-1.5) GO TO 909
C!!!  SPECIFY TANGENT VECTORS
      TX1=BTS(1)
      TY1=BTS(2)
      TX2=BTS(3)
      TY2=BTS(4)
      ER(1)=TX1*CPS+TY1*SPS
      ER(2)=TX2*CPS+TY2*SPS
C!!!  LOOP THRU TANGENT VECTORS
      I=1
      LRFCT=.FALSE.
      VI(1)=VTS(1)
      VI(2)=VTS(2)
      IF(LDEBUG) WRITE(6,900)
  900 FORMAT(/,' DEBUGGING SCTCYL SUBROUTINE')
    3 CONTINUE
      CALL NANDB(UN,UB,VI(I))
      SINA=UN(1)*CPS+UN(2)*SPS
C!!!  CALCULATE ALR, THE PROPAGATION DIRECTION PHI ANGLE
C!!!  IN TAN POINT COORDINATE SYSTEM.
      ALR=BTAN2(SINA,-ER(I))
      IF(ALR.LT.0.) ALR=ALR+TPI
C!!!  IF GRAZING INCIDENCE IS PRESENT, SKIP TO APPROPRIATE SECTION
      IF(ABS(PI-ALR).LT.0.0085) GO TO 5
C!!!  IF ALR IS G.T. THAN PI, COMPUTE CREEPING WAVE TERMS
      IF(ALR.GT.PI) GO TO 10
C!!!  IF ALR .LE. PI, COMPUTE G.O. RAY PATH AND FIELD
C!!!  COMPONENTS
      CALL REFCYL(ERT,ERP)
      IF(LTRF) GO TO 1
      SNAS=UN(1)*XS(1)+UN(2)*XS(2)
      IC=2*I-1
      CSAS=BTS(IC)*XS(1)+BTS(IC+1)*XS(2)
      ALS=BTAN2(SNAS,-CSAS)
      ALRS=ALR-ALS
C!!!  IS REFLECTION TO BE HANDLED WITH OTHER TAN VECTOR?
      IF(ABS(ALRS).LT.0.0085.AND.I.EQ.2) GO TO 1
      IF(ALRS.LE.-0.0085) GO TO 1
      GM=(PI*RG)**(1./3.)
```

```
        RHS=RHO1*DL/(DL-RHO1)
        SKWIG=-ABS(2.*TPI*RHS/GM/GM)
        CF=-SQRT(-2./PI/SKWIG)*CPI4*REF
        CF=CF*CEXP(-CJ*SKWIG*SKWIG*SKWIG/12.)
        TTRM=SKWIG/GM
        XX=PI*(DL+RHS)*TTRM*TTRM
C!!!    SPECIFY HARD AND SOFT COMPONENTS OF FIELD
C!!!    INCIDENT ON CYLINDER (FROM REFCYL)
        EST=ESTH
        ESP=ESPH
        EHT=EHTH
        EHP=EHPH
        GO TO 30
   10   CONTINUE
        IF(LRFCT) LRFC=.FALSE.
        LRFCT=.TRUE.
C!!!    COMPUTE CREEPING WAVE TERMS IF ALR .GT. PI
C!!!    COMPUTE INCIDENT RAY TANGENT POINT
        XI=A*COS(VI(I))
        YI=B*SIN(VI(I))
        XD=XI-XS(1)
        YD=YI-XS(2)
        S=SQRT(XD*XD+YD*YD)
        ZI=S*CTHS/STHS+XS(3)
C!!!    IS TANGENT POINT OFF OF FINITE CYLINDER?
        IF(ZI.GT.ZC(1)+XI*CTC(1).OR.
       2ZI.LT.ZC(2)+XI*CTC(2)) GO TO 1
        ZD=ZI-XS(3)
        PHIR=BTAN2(YD,XD)
        THIR=BTAN2(S,ZD)
        S=SQRT(S*S+ZD*ZD)
        DI(1)=XD/S
        DI(2)=YD/S
        DI(3)=ZD/S
C!!!    DOES INCIDENT RAY HIT PLATE BEFORE CYLINDER?
        CALL PLAINT(XS,DI,DHIT,0,LHIT)
        IF(LHIT.AND.(DHIT.LT.S)) GO TO 1
C!!!    CALCULATE INCIDENT FIELD PATTERN FACTOR
        CALL SOURCE(EF,EG,EIX,EIY,EIZ,THIR,PHIR,VXS)
        IF(LDEBUG) WRITE(6,*) EF,EG
C!!!    PERFORM CREEPING WAVE COMPUTATIONS
        IF(I.EQ.1) VD=BTAN2(-B*CPS,A*SPS)
        IF(I.EQ.2) VD=BTAN2(B*CPS,-A*SPS)
        VDP=VD-VI(I)
        IF(VDP.GT.PI) VDP=VDP-TPI
        IF(VDP.LT.-PI) VDP=VDP+TPI
        IF(I.EQ.2) GO TO 20
        IF(VDP.LT.0.) GO TO 1
        VL=VI(I)
        VU=VDP+VI(I)
        GO TO 25
   20   CONTINUE
        IF(VDP.GT.0.) GO TO 1
        VL=VDP+VI(I)
        VU=VI(I)
   25   CONTINUE
        CALL FKARG(SKWIG,AS,VL,VU)
        XRF(1)=A*COS(VD)
        XRF(2)=B*SIN(VD)
        ID=3
        CALL DCG32(VL,VU,FCT,SS)
        SS=SS/SAS
        XRF(3)=ZI+SS*CTHS
C!!!    DOES RAY HIT PLATE AFTER LEAVING CYLINDER?
        CALL PLAINT(XRF,D,DHIT,P,LHIT)
        IF(LHIT) GO TO 1
```

```
      CALL RADCV(RGI,RT,VI(I))
      CALL RADCV(RGF,RT,VD)
      GMM=(PI*PI*RGI*RGF)**(1./6.)
      CF=-GMM*CPI4*CEXP(-CJ*TPI*(S+SS))/PI/SQRT(2.*S)
C!!!  COMPUTE PHASE TERM
      CF=CF*CEXP(CJ*TPI*(XRF(1)*D(1)+XRF(2)*D(2)+XRF(3)*D(3)))
      TTRM=SKMIG/GMM
      XX=PI*S*TTRM*TTRM
      BX=-UN(2)*DI(3)
      BY=UN(1)*DI(3)
      BZ=UN(2)*DI(1)-UN(1)*DI(2)
      ESP=(0.,0.)
      EHT=(0.,0.)
C!!!  COMPUTE HARD AND SOFT CREEPING WAVE COMPONENTS
      EHP=EIX*UN(1)+EIY*UN(2)
      EST=EIX*BX+EIY*BY+EIZ*BZ
      IF(I.EQ.1) EHP=-EHP
      IF(I.EQ.2) EST=-EST
30    CONTINUE
C!!!  COMPUTE THE TRANSITION FIELD
      XXS=SQRT(TPI*XX)
      XXX=SQRT(2.*XX/PI)
      CALL FRNELS(CCC,SSS,XXX)
      FI=CMPLX(0.5-CCC,SSS-0.5)
      FI=XXS*FI*CEXP(CJ*(.5*PI+XX))
      FI=-FI/SKMIG/SQRT(2.)
      SQTP=SQRT(2.*PI)
      CFH=CF*(FI+SQTP*QFUN(SKMIG))
      CFS=CF*(FI+SQTP*PFUN(SKMIG))
      DEPH=CFH*EHP+CFS*ESP
      DETH=CFH*EHT+CFS*EST
      GO TO 6
5     CONTINUE
C!!!  COMPUTE GRAZING INCIDENCE TRANSITION FIELD
C!!!  DOES RAY HIT PLATE?
      CALL PLAINT(XS,D,DHIT,0,LHIT)
      IF(LHIT) GO TO 1
      SGN=-SIGN(1.,SIN(ALR))
C!!!  CALCULATE INCIDENT POINT
      XI=A*COS(VI(I))
      YI=B*SIN(VI(I))
      XD=XI-XS(1)
      YD=YI-XS(2)
      S=SQRT(XD*XD+YD*YD)
      ZI=S*CTHS/STHS+XS(3)
C!!!  IS POINT OF GRAZING INCIDENCE OFF OF FINITE CYLINDER?
      IF(ZI.GT.ZC(1)+XI*CTC(1).OR.
     2ZI.LT.ZC(2)+XI*CTC(2)) GO TO 1
      ZD=ZI-XS(3)
      S=SQRT(S*S+ZD*ZD)
      CALL RADCV(RGI,RT,VI(I))
      GM=(PI*RGI)**(1./3.)
C!!!  CALCULATE INCIDENT FIELD PATTERN FACTOR
      CALL SOURCE(EF,EG,EIX,EIY,EIZ,THSR,PHSR,VXS)
C!!!  CALCULATE PHASE TERM
      CF=CEXP(CJ*TPI*(XS(1)*D(1)+XS(2)*D(2)+XS(3)*D(3)))
      BX=-UN(2)*D(3)
      BY=UN(1)*D(3)
      BZ=UN(2)*D(1)-UN(1)*D(2)
C!!!  CALCULATE HARD AND SOFT COMPONENTS OF INCIDENT FIELD
      EHP=EIX*UN(1)+EIY*UN(2)
      EST=EIX*BX+EIY*BY+EIZ*BZ
      IF(I.EQ.1) EHP=-EHP
      IF(I.EQ.2) EST=-EST
      CFH=GM*CPI4*QFUN(0.)/SQRT(PI*S)
      CFS=GM*CPI4*PFUN(0.)/SQRT(PI*S)
```

```
C!!!  CALCULATE TOTAL GRAZING INCIDENCE FIELD
      DETH=(0.5*EF*SGN-CFS*EST)*CF
      DEPH=(0.5*EG*SGN-CFH*EHP)*CF
6     CONTINUE
C!!!  CALCULATE TOTAL CYLINDER FIELDS
      EP=EP+DEPH
      ET=ET+DETH
      IF(LDEBUG) WRITE(6,*) I,SKMIG,DX,FI,CF
      IF(LDEBUG) WRITE(6,*) CFH,CFS
      IF(LDEBUG) WRITE(6,*) EHT,EST
      IF(LDEBUG) WRITE(6,*) EHP,ESP
      IF(LDEBUG) WRITE(6,*) DETH,DEPH
1     I=I+1
      IF(I.LE.2) GO TO 3
909   CONTINUE
      IF(.NOT.LTEST) RETURN
      WRITE(6,910)
910   FORMAT(/,' TESTING SCTCYL SUBROUTINE')
      WRITE(6,*) ET,EP
      WRITE(6,*) ERT,ERP
      RETURN
      END
```

## SOURCE

### PURPOSE

To compute the source field pattern factor for radiation in a given direction from the sourre.

### PERTINENT GEOMETRY

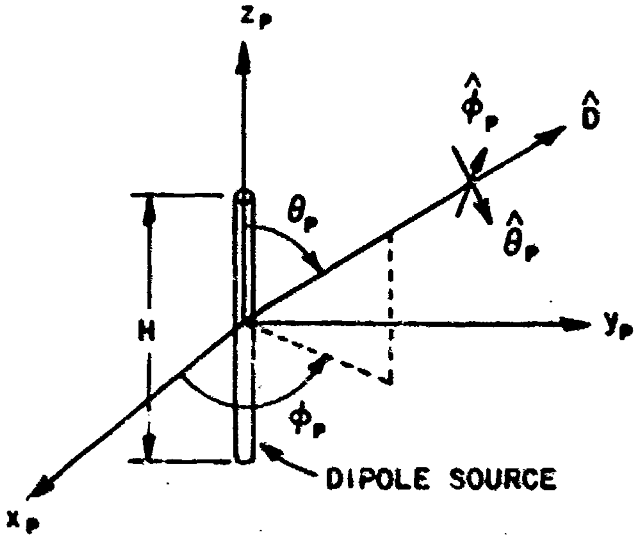


Figure 107--Illustration of one dimensional source (dipole)

Note - one dimensional source always along $z_p$ axis

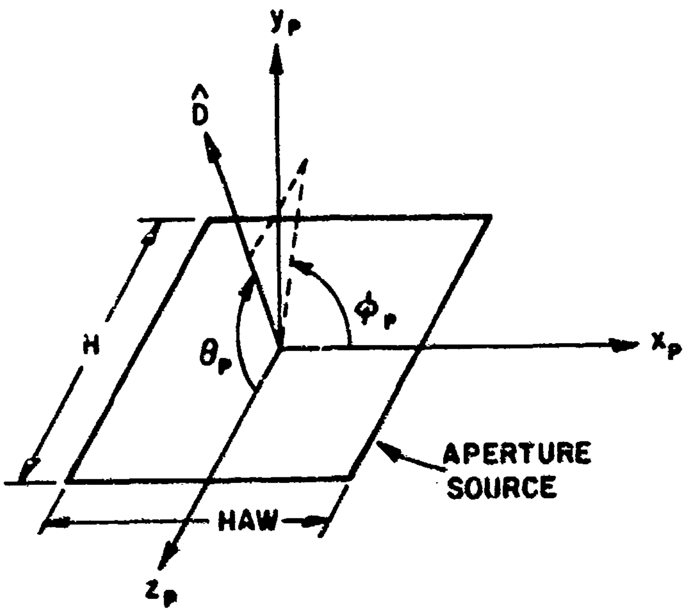


Figure 108--Illustration of two dimensional (aperture) source.

Note - two dimensional source always in $x_p$-$z_p$ plane with current in the $\hat{z}_p$ direction.

METHOD

The source distribution is given as follows

line source: $$\left.\begin{matrix} I(z_p) \\ K(z_p) \end{matrix}\right\} = \left\{\begin{matrix} I_m \\ K_m \end{matrix}\right\} \cos \frac{\pi z_p}{H} \quad x_p=0 \; y_p=0, \; \frac{-H}{2} \le z_p \le \frac{H}{2}$$

aperture source: $$\left.\begin{matrix} J(z_p,x_p) \\ M(z_p,x_p) \end{matrix}\right\} = \left\{\begin{matrix} J_m \\ M_m \end{matrix}\right\} \cos \frac{\pi z_p}{H} \quad y_p=0, \frac{-HAW}{2} \le x_p \le \frac{HAW}{2}$$

$$\frac{-H}{2} \le z_y \le \frac{H}{2}$$

where $x_p$, $y_p$, $z_p$ are unit vectors of the source coordinate systems

$$\hat{x}_p = \hat{x}\,VAX(1,1) + \hat{y}\,VAX(1,2) + \hat{z}\,VAX(1,3)$$
$$\hat{y}_p = \hat{x}\,VAX(2,1) + \hat{y}\,VAX(2,2) = \hat{z}\,VAX(2,3)$$
$$\hat{z}_p = \hat{x}\,VAX(3,1) + \hat{y}\,VAX(3,2) + \hat{z}\,VAX(3,3).$$

The far-zone electric field is given by

$$\bar{E}(\theta_p,\phi_p) = \bar{E}_o F_z(\theta_p) F_x(\theta_p,\phi_p) \frac{e^{-jks'}}{s'} .$$

where for an electric source,

$$\bar{E}_o \begin{cases} \hat{\theta}_p \frac{j\eta}{\pi} I_m H, & \text{line source} \\ \hat{\theta}_p \frac{j\eta}{\pi} J_m H \; HAW, & \text{aperture source} \end{cases}$$

and for a magnetic source,

$$\bar{E}_o = \begin{cases} -\hat{\phi}_p \frac{j}{\pi} K_m H, & \text{line source} \\ -\hat{\phi}_p \frac{j}{\pi} M_m H \; HAW, & \text{aperture source} \end{cases}$$

and where

$$F_z(\theta_p) = \frac{\sin\theta_p \cos(\pi H \cos\theta_p)}{(1-4H^2 \cos^2\theta_p)}$$

$$F_x(\theta_p,\phi_p) = \begin{cases} 1 & \text{, line source} \\ \dfrac{\sin(\pi \text{ HAW } \sin\theta_p \cos\phi_p)}{\pi \text{ HAW } \sin\theta_p \cos\phi_p} & \text{, aperture source.} \end{cases}$$

Note that all diagrams and formulae on this and the preceeding page refer to the source coordinate system. The subroutine returns the field components in the reference coordinate system.

The far-zone E-field radiated by the source is then given in the reference coordinate system by

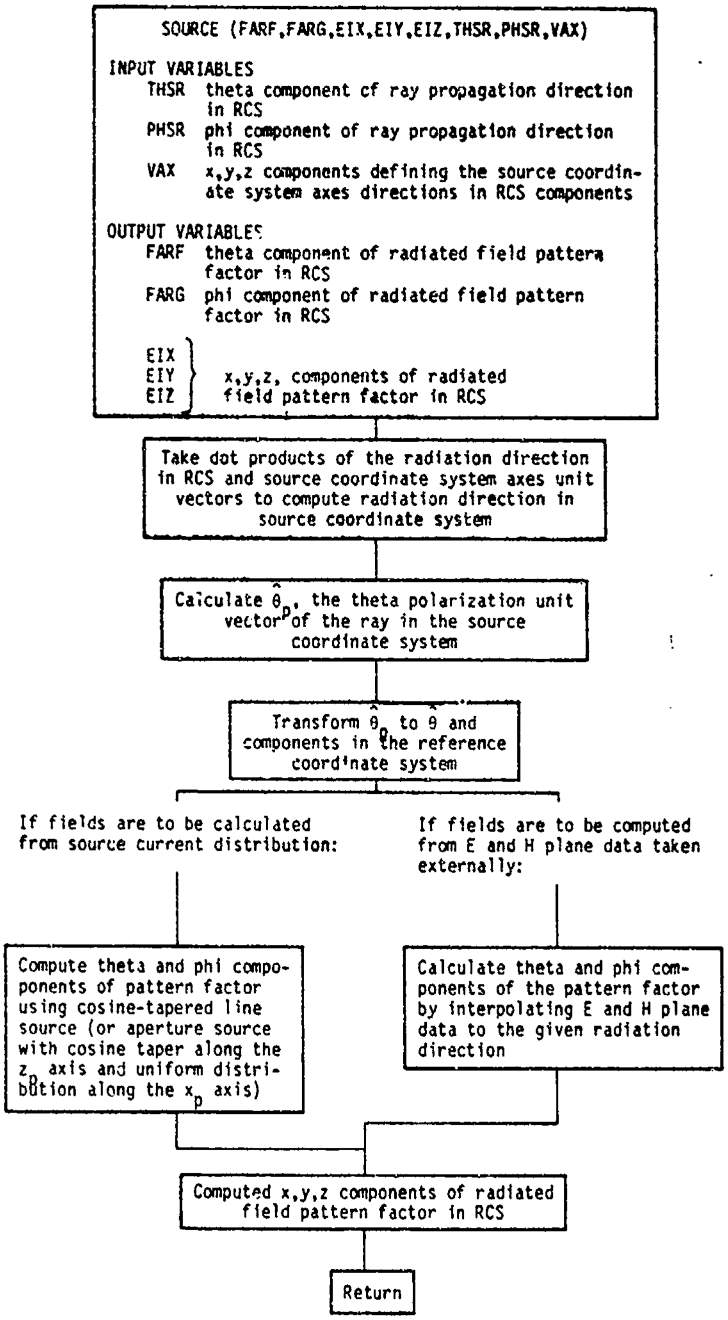
$$\overline{E}(r,\theta,\phi) = W_m(\text{FARF}\hat{\theta}+\text{FARG}\hat{\phi}) \frac{e^{-jkR}}{R}$$

or

$$\overline{E}(x,y,z) = W_m(\text{EIX } \hat{x} + \text{EIY } \hat{y} + \text{EIZ } \hat{z}) \frac{e^{-jkR}}{R}$$

Note that the factor $\frac{e^{-jkR}}{R}$ and the source weights ($W_m$=$I_m$,$K_m$,$J_m$, or $M_m$) are not included in subroutine SOURCE, but are added elsewhere in the code. Note also that the interpolation fields are not fully implemented in this version of the code.

# FLOW DIAGRAM



SOURCE (FARF,FARG,EIX,EIY,EIZ,THSR,PHSR,VAX)

INPUT VARIABLES
THSR theta component cf ray propagation direction in RCS
PHSR phi component of ray propagation direction in RCS
VAX x,y,z components defining the source coordinate system axes directions in RCS components

OUTPUT VARIABLES
FARF theta component of radiated field pattern factor in RCS
FARG phi component of radiated field pattern factor in RCS

EIX
EIY } x,y,z, components of radiated field pattern factor in RCS
EIZ

Take dot products of the radiation direction in RCS and source coordinate system axes unit vectors to compute radiation direction in source coordinate system

Calculate $\hat{\theta}_p$, the theta polarization unit vector of the ray in the source coordinate system

Transform $\hat{\theta}_p$ to $\hat{\theta}$ and components in the reference coordinate system

If fields are to be calculated from source current distribution:

Compute theta and phi compoponents of pattern factor using cosine-tapered line source (or aperture source with cosine taper along the $z_p$ axis and uniform distribution along the $x_p$ axis)

If fields are to be computed from E and H plane data taken externally:

Calculate theta and phi components of the pattern factor by interpolating E and H plane data to the given radiation direction

Computed x,y,z components of radiated field pattern factor in RCS

Return

## SYMBOL DICTIONARY

| | |
|---|---|
| AWFAC | PATTERN FACTOR (FX) OF SOURCE FIELD DUE TO XP DIMENSION OF APERTURE |
| BF | INTERPOLATION VARIABLE |
| CPHP | COSINE OF PHP |
| CTHP | COSINE OF THP |
| EFD | INTERPOLATED E FIELD |
| EFED | E PLANE SOURCE FIELD PATTERN MEASURED VALUES |
| EX | COMPUTATIONAL VARIABLE |
| EXI | PATTERN FACTOR FZ |
| F | DOT PRODUCT OF THETA UNIT POLARIZATION VECTOR OF SOURCE COORD SYS AND THETA UNIT VECTOR OF RCS |
| FW | ARGUMENT OF PATTERN FACTOR FX |
| G | DOT PRODUCT OF THETA UNIT POLARIZATION VECTOR OF SOURCE COORD SYS AND PHI UNIT VECTOR OF RCS |
| HFD | INTERPOLATED H FIELD |
| HFED | H PLANE SOURCE PATTERN MEASURED VALUES |
| IT | INTERPOLATION VARIABLE |
| PHP | PHI COMPONENT OF RADIATION DIRECTION IN SOURCE COORDINATE SYSTEM |
| PHSR | PHI COMPONENT OF RADIATION DIRECTION IN RCS |
| RDX | DOT PRODUCT OF RADIATION DIRECTION AND XP AXIS UNIT VECTOR |
| RDY | DOT PRODUCT OF RADIATION DIRECTION AND YP AXIS UNIT VECTOR |
| SPHP | SINE OF PHP |
| THP | THETA COMPONENT OF RADIATION DIRECTION IN SOURCE COORDINATE SYSTEM |
| THSR | THETA COMPONENT OF RADIATION DIRECTION IN RCS |
| VAX | X,Y,Z COMPONENTS DEFINING AXES OF SOURCE (OR SOURCE IMAGE) COORDINATE SYSTEM |
| XTH<br>YTH<br>ZTH | X,Y,Z COMPONENTS OF THE THETA POLARIZATION UNIT VECTOR OF THE RAY IN THE SOURCE COORDINATE SYSTEM (IN RCS COMPONENTS) |

# CODE LISTING

```
C-----------------------------------------------------------------------
      SUBROUTINE SOURCE(FARF,FARG,EIX,EIY,EIZ,THSR,PHSR,VAX)
C!!!
C!!!  SOURCE FIELD
C!!!
      COMPLEX EX,EIX,EIY,EIZ,FARF,FARG
      COMPLEX EFED(1),HFED(1),EFD,HFD
      DIMENSION VAX(3,3)
      LOGICAL LSOR
      COMMON/FARP/IM,H,HAW
      COMMON/PIS/PI,TPI,DPR,RPD
      COMMON/SOURSF/FACTOR
      COMMON/FEDDAT/EFED,HFED
      CTHS=COS(THSR)
      STHS=SIN(THSR)
      CPHS=COS(PHSR)
      SPHS=SIN(PHSR)
C!!!  TAKE DOT PRODUCTS OF THE RADIATION DIRECTION UNIT
C!!!  VECTOR AND SOURCE COORD SYS (PRIMED)AXES
C!!!  UNIT VECTORS TO OBTAIN THP AND PHP (PROPAGATION
C!!!  ANGLES IN THE SOURCE COORD SYSTEM)
      CTHP=VAX(3,1)*CPHS*STHS+VAX(3,2)*SPHS*STHS+VAX(3,3)*CTHS
      RDX=VAX(1,1)*CPHS*STHS+VAX(1,2)*SPHS*STHS+VAX(1,3)*CTHS
      RDY=VAX(2,1)*CPHS*STHS+VAX(2,2)*SPHS*STHS+VAX(2,3)*CTHS
      STHP=SQRT(RDX*RDX+RDY*RDY)
      CPHP=RDX/STHP
      SPHP=RDY/STHP
C!!!  CALCULATE THETA POLARIZATION UNIT VECTOR FOR RAY
C!!!  IN SOURCE COORD SYS AND REPRESENT WITH X,Y,Z
C!!!  COMPONENTS IN THE REFERENCE COORDINATE SYSTEM
      XTH=VAX(1,1)*CPHP*CTHP+VAX(2,1)*SPHP*CTHP-VAX(3,1)*STHP
      YTH=VAX(1,2)*CPHP*CTHP+VAX(2,2)*SPHP*CTHP-VAX(3,2)*STHP
      ZTH=VAX(1,3)*CPHP*CTHP+VAX(2,3)*SPHP*CTHP-VAX(3,3)*STHP
C!!!  TRANSFORM THETA POLARIZATION UNIT VECTOR TO
C!!!  RCS COMPONENTS
      F=XTH*CTHS*CPHS+YTH*CTHS*SPHS-ZTH*STHS
      G=-XTH*SPHS+YTH*CPHS
      IF(IM.EQ.3) GO TO 10
C!!!  CALCULATE FIELDS USING COSINE TAPERED LINE SOURCE
C!!!  (OR APERTURE SOURCE WITH COSINE TAPER IN ZP DIRECTION
C!!!  AND UNIFORM DISTRIBUTION IN THE XP DIRECTION)
      EX1=STHP
      ACTHP=ABS(CTHP)
      IF(ABS(ACTHP-.5/H).LT.1.E-5) GO TO 5
      EX1=2.*H*STHP*COS(PI*H*CTHP)/(1.-4.*H*H*CTHP*CTHP)
      GO TO 6
    5 EX1=.25*PI*SQRT(4.*H*H-1.)
    6 CONTINUE
      AWFAC=1.0
      IF(HAW.LT.0.1) GO TO 7
      FW=PI*HAW*STHP*CPHP
      IF(ABS(FW).LT.1.E-05) FW=1.E-05
      AWFAC=HAW*SIN(FW)/FW
    7 EX1=EX1*AWFAC
      EX=CMPLX(0.,EX1*FACTOR)
      FARF=F*EX*60.
      FARG=G*EX*60.
C!!!  USE DUALITY FOR MAGNETIC CURRENT SOURCE
      IF(IM.EQ.1)FARG=-F*EX/TPI
      IF(IM.EQ.1)FARF=G*EX/TPI
      GO TO 20
   10 CONTINUE
C!!!  CALCULATE FIELDS BY INTERPOLATION E AND H-PLANE DATA
C!!!  (TAKEN EXTERNALLY) TO THE GIVEN RADIATION DIRECTION
```

```
      CTHF=SPHP*STHP
      BF=CPHP*CPHP*STHP*STHP+CTHP*CTHP
      STHF=SQRT(BF)
      THF=DPR*BTAN2(STHF,CTHF)
      ITF=THF
      IT=ITF+1
      EFD=EFED(IT)+(EFED(IT+1)-EFED(IT))*(THF-ITF)
      HFD=HFED(IT)+(HFED(IT+1)-HFED(IT))*(THF-ITF)
      IF(ABS(BF).LT.1.E-3) GO TO 15
      EX=EFD*CPHP*CPHP*STHP*STHP+HFD*CTHP*CTHP
      EX=EX/BF
      GO TO 16
 15   EX=EFD
 16   CONTINUE
      FARG=-F*EX
      FARF=G*EX
 20   CONTINUE
C!!!  COMPUTE X,Y,Z COMPONENTS OF SOURCE PATTERN FACTOR
      EIX=FARF*CTHS*CPHS-FARG*SPHS
      EIY=FARF*CTHS*SPHS+FARG*CPHS
      EIZ=-FARF*STHS
      RETURN
      END
```

## SOURCP

### PURPOSE

To compute the normal derivative, $\frac{\partial \overline{E}^i}{\partial n}$, of the incident field pattern factor for source ray incident on a given edge (to be used in slope diffraction computation).

### PERTINENT GEOMETRY

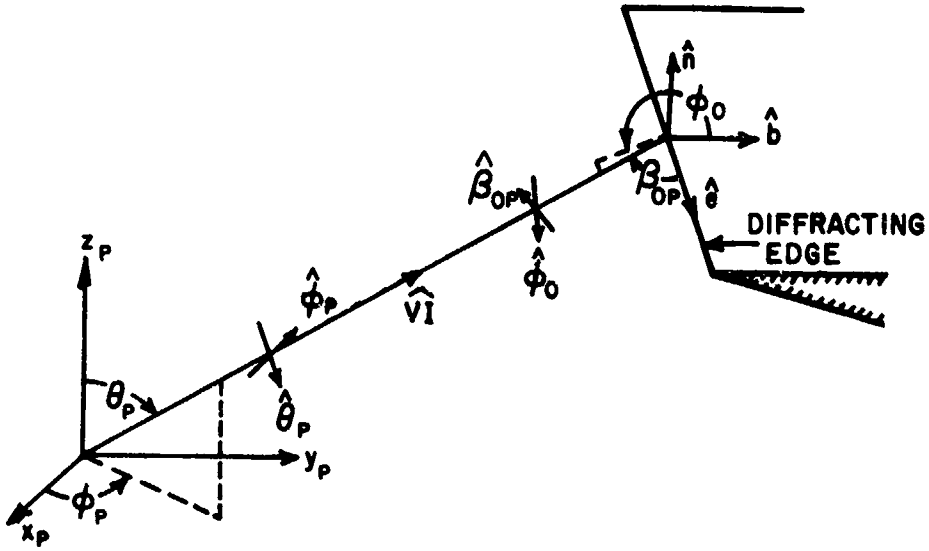


Figure 109--Geometry of source field incident on plate edge.

### METHOD

The slope field is given as follows:

$$\frac{\partial}{\partial n}\overline{E}^i = \frac{1}{s' \sin\beta_o} \frac{\partial}{\partial \phi_o} \overline{E}^i$$

where

$$\overline{E}^i = \overline{E}_o \underbrace{\frac{\sin\theta_p \cos(\pi H \cos\theta_p)}{(1-4H^2 \cos^2\theta_p)}}_{F_z(\theta_p)} \underbrace{\frac{\sin(\pi \text{ HAW } \sin\theta_p \cos\phi_p)}{\pi \text{ HAW } \sin\theta_p \cos\phi_p}}_{F_x(\theta_p,\phi_p)} \frac{e^{-jks'}}{s'}$$

For electric source

$$\overline{E}_o = \begin{cases} \hat{\theta}_p \frac{j\eta}{\pi} I_m H \text{ , line source} \\ \hat{\theta}_p \frac{j\eta}{\pi} J_m \text{ HAW H , aperture source} \end{cases}$$

$$\overline{E}^i = \overline{E}_o \, F_z(\theta_p) F_x(\theta_p,\phi_p) \frac{e^{-jks'}}{s'} = E_{\theta p} \, \hat{\theta}_p$$

$$\frac{\partial \overline{E}^i}{\partial \phi_o} = \frac{\partial (E_{\theta p} \, \hat{\theta}_p)}{\partial \phi_o} = \frac{\partial E_{\theta p}}{\partial \phi_o} \, \hat{\theta}_p + E_{\theta p} \frac{\partial \hat{\theta}_p}{\partial \phi_o}$$

$$\frac{\partial E_{\theta p}}{\partial \phi_o} = \frac{\partial E_{\theta p}}{\partial \theta_p} \frac{\partial \theta_p}{\partial \phi_o} + \frac{\partial E_{\theta p}}{\partial \phi_p} \frac{\partial \phi_p}{\partial \phi_o}$$

$$\frac{\partial E_{\theta p}}{\partial \theta_p} = E_o \left( \frac{\partial F_z(\theta_p)}{\partial \theta_p} \, F_x(\theta_p,\phi_p) + F_z \frac{\partial F_x(\theta_p,\phi_p)}{\partial \theta_p} \right) \frac{e^{-jks'}}{s'} ,$$

$$\frac{\partial E_{\theta p}}{\partial \phi_p} = E_o \left( F_z(\theta_p) \frac{\partial F_x(\theta_p,\phi_p)}{\partial \phi_p} \right) \frac{e^{-jks'}}{s'}$$

$$F_z(\theta_p) = \frac{\sin\theta_p \cos(\pi H \cos\theta_p)}{(1 - 4H^2 \cos^2\theta_p)}$$

$$\frac{\partial F_z}{\partial \theta_p} = \{ [(1-4H^2\cos^2\theta_p)(\cos\theta_p \cos(\pi H \cos\theta_p) + \sin^2\theta_p \pi H \sin(\pi H \cos\theta_p)]$$

$$+ [-8H^2 \cos\theta_p \sin^2\theta_p \cos(\pi H \cos\theta_p)]\} \frac{1}{(1 - 4H^2\cos^2\theta_p)^2}$$

$$F_x = \frac{\sin(\pi \text{ HAW } \sin\theta_p \cos\phi_p)}{\pi \text{ HAW } \sin\theta_p \cos\phi_p}$$

$$\frac{\partial F_x}{\partial \theta_p} = \cot\theta_p \left[ \cos(\pi \text{ HAW } \sin\theta_p \cos\phi_p) - \frac{\sin(\pi \text{ HAW } \sin\theta_p \cos\phi_p)}{\pi \text{ HAW } \sin\theta_p \cos\phi_p} \right]$$

$$\frac{\partial F_x}{\partial \phi_p} = \tan\phi_p \left[ \frac{\sin(\pi \text{ HAW } \sin\theta_p \cos\phi_p)}{\pi \text{ HAW } \sin\theta_p \cos\phi_p} - \cos(\pi \text{ HAW } \sin\theta_p \cos\phi_p) \right]$$

$$\frac{\partial \theta_p}{\partial \phi_o} = -\sin\beta_{op} \hat{\phi}_o \cdot \hat{\theta}_p$$

$$\frac{\partial \phi_p}{\partial \phi_o} = -\frac{\sin\beta_{op}}{\sin\theta_p} \hat{\phi}_o \cdot \hat{\phi}_p$$

$$\frac{\partial \hat{\theta}_p}{\partial \phi_o} = \sin\beta_{op} [\hat{\phi}_o \cdot \hat{\theta}_p \ \widehat{VI} - \cot\theta_p \ \hat{\phi}_o \cdot \hat{\phi}_p \ \hat{\phi}_p]$$

$$\frac{\partial \hat{\phi}_p}{\partial \phi_o} = \frac{\sin\beta_{op}}{\sin\theta_p} (\hat{\phi}_o \cdot \hat{\phi}_p) \hat{\rho}_p$$

$$\hat{\rho}_p = \sin\theta_p \ \widehat{VI} + \cos\theta_p \ \hat{\theta}_p$$

$$\hat{\theta}_p = \hat{x} \text{ XTH} + \hat{y} \text{ YTH} + \hat{z} \text{ ZTH}$$

$$\hat{\phi}_p = \hat{x} \text{ XPH} + \hat{y} \text{ YPH} + \hat{z} \text{ ZPH}$$

combining,

$$\frac{\partial \bar{E}^i}{\partial n} = \frac{j\eta H}{\pi} \begin{Bmatrix} I_m \\ J_m \, HAW \end{Bmatrix} \hat{\phi}_o \cdot \Big[ F_x F_z \hat{\theta}_p \hat{V}I - \left( \frac{\partial F_z}{\partial \theta_p} F_x + F_z \frac{\partial F_x}{\partial \theta_p} \right) \hat{\theta}_p \hat{\theta}_p -$$

$$\frac{1}{\sin\theta_p} F_z \frac{\partial F_x}{\partial \phi_p} \hat{\phi}_p \hat{\theta}_p - \cot\theta_p F_x F_z \hat{\phi}_p \hat{\phi}_p \Big] \frac{e^{-jks'}}{s'^2} \quad .$$

The slope fields for a magnetic source are derived in a similar manner yielding

$$\frac{\partial \bar{E}^i}{\partial n} = \frac{-j}{\pi} H \begin{Bmatrix} K_m \\ M_m HAW \end{Bmatrix} \hat{\phi}_o \cdot \Big[ F_x F_z \hat{\phi}_p \hat{V}I - \left( \frac{\partial F_z}{\partial \theta_p} F_x + F_z \frac{\partial F_x}{\partial \theta_p} \right) \hat{\theta}_p \hat{\phi}_p -$$

$$\frac{1}{\sin\theta_p} F_z \frac{\partial F_x}{\partial \phi_p} \hat{\phi}_p \hat{\phi}_p + \cot\theta_p F_x F_z \hat{\phi}_p \hat{\theta}_p \Big] \frac{e^{-jks'}}{s'^2} \quad .$$

The normal derivative of the incident field, $\frac{\partial \bar{E}^i}{\partial n}$, is returned in components perpendicular and parallel to the edge (referred to as hard and soft components):

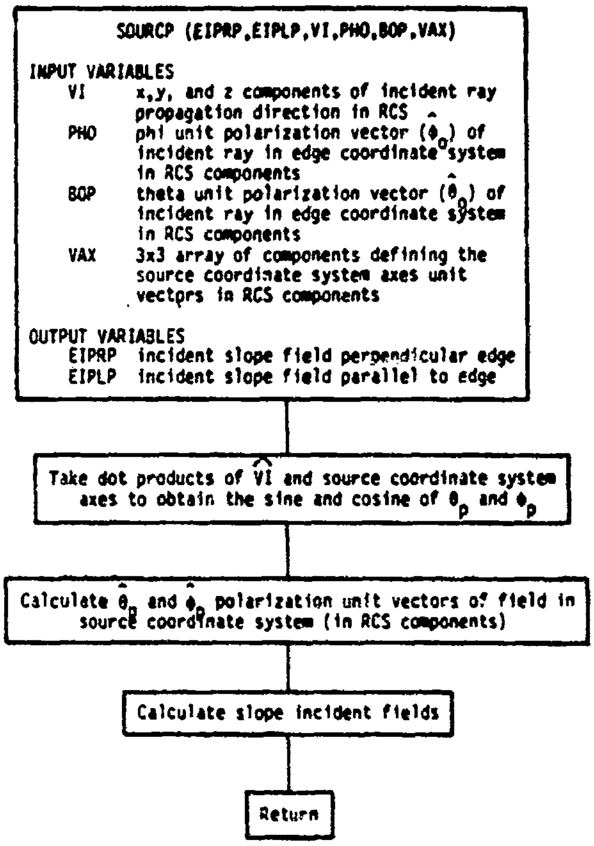
$$\frac{\partial \bar{E}^i}{\partial n} = \left( \frac{\partial \bar{E}^i}{\partial n} \cdot \hat{\phi}_o \right) \hat{\phi}_o + \left( \frac{\partial \bar{E}^i}{\partial n} \cdot \hat{\beta}_{op} \right) \hat{\beta}_{op}$$

Acoustically hard case    Acoustically soft case

$$\frac{\partial \bar{E}^i}{\partial n} = W_m \Big[ EIPRP \; \hat{\phi}_o + EIPLP \; \hat{\beta}_{op} \Big] \frac{e^{-jks'}}{s'^2}$$

Note that the factors $\frac{e^{-jks'}}{s'^2}$, along with the source weights ($W_m = I_m$, $K_m, J_m$, or $M_m$) are added elsewhere in the code.

## FLOW DIAGRAM


SOURCP (EIPRP,EIPLP,VI,PHO,BOP,VAX)
INPUT VARIABLES
VI x,y, and z components of incident ray propagation direction in RCS
PHO phi unit polarization vector ($\hat{\phi}_o$) of incident ray in edge coordinate system in RCS components
BOP theta unit polarization vector ($\hat{\theta}_o$) of incident ray in edge coordinate system in RCS components
VAX 3x3 array of components defining the source coordinate system axes unit vectors in RCS components
OUTPUT VARIABLES
EIPRP incident slope field perpendicular edge
EIPLP incident slope field parallel to edge
Take dot products of $\widehat{VI}$ and source coordinate system axes to obtain the sine and cosine of $\theta_p$ and $\phi_p$
Calculate $\hat{\theta}_p$ and $\hat{\phi}_p$ polarization unit vectors of field in source coordinate system (in RCS components)
Calculate slope incident fields
Return

# SYMBOL DICTIONARY

| | |
|---|---|
| ACTHP | ABSOLUTE VALUE OF CTHP |
| ARG | ARGUMENT OF FX |
| BOP | X,Y,Z COMPONENTS OF BETA POLARIZATION UNIT VECTOR FOR RAY INCIDENT ON EDGE (EDGE-CENTERED COORD SYS) |
| CPHP | COS(PHP) |
| CTHP | COS(THP) |
| E1 | COMPUTATIONAL VARIABLE |
| E2 | COMPUTATIONAL VARIABLE |
| EA | COMPUTATIONAL VARIABLE |
| EB | COMPUTATIONAL VARIABLE |
| EFA | PARTIAL DERIVATIVE OF FZ WITH THP |
| EFB | FZ DIVIDED BY SIN(THP) |
| EFC | PARTIAL OF FX WITH THP DIVIDED BY COT(THP) |
| EFD | PARTIAL OF FX WITH PHP |
| EFE | FX TIMES HAW |
| EFF | PARTIAL OF FP WITH THP |
| EIPLP | SOFT COMPONENT OF THE SLOPE FIELDS |
| EIPHP | HARD COMPONENT OF THE SLOPE FIELDS |
| PHO | X,Y,Z COMPONENTS OF PHI POLARIZATION UNIT VECTOR FOR RAY INCIDENT ON EDGE (EDGE-CENTERED COORD SYS) |
| PHP | PHI COMPONENT OF PROPAGATION DIRECTION IN SOURCE COORD SYS |
| PPBO | DOT PRODUCT OF PHI POL UNIT VECTOR OF SOURCE COORD SYS AND BETA POL UNIT VECTOR OF EDGE-CENTERED COORD SYS |
| PPHO | DOT PRODUCT OF THE PHI POLARIZATION UNIT VECTOR OF THE SOURCE COORD SYS AND THE PHI UNIT POLARIZATION VECTOR OF THE EDGE-CENTERED COORDINATE SYSTEM |
| RDX | DOT PRODUCT OF VI AND XP AXIS UNIT VECTOR |
| RDY | DOT PRODUCT OF VI AND YP AXIS UNIT VECTOR |
| SN | SIGN OF COS(THP) |
| SNARG | SIN(ARG)/ARG |
| SPHP | SIN(PHP) |
| STHP | SIN(THP) |
| THP | THETA COMPONENT OF THE PROPAGATION DIRECTION IN THE SOURCE COORDINATE SYSTEM |
| TPBO | DOT PRODUCT OF THE THETA POLARIZATION UNIT VECTOR OF THE SOURCE COORDINATE SYSTEM AND THE BETA POLARIZATION UNIT VECTOR OF THE EDGE-CENTERED COORDINATE SYSTEM |
| TPHO | DOT PRODUCT OF THE THETA POLARIZATION UNIT VECTOR OF THE SOURCE COORDINATE SYSTEM AND THE PHI POLARIZATION UNIT VECTOR OF THE EDGE-CENTERED COORDINATE SYSTEM |
| VI | X,Y,Z COMPONENTS OF THE RAY PROPAGATION DIRECTION IN RCS |
| XPH<br>YPH<br>ZPH | X,Y,Z COMPONENTS OF THE PHI UNIT POLARIZATION VECTOR OF THE FIELD IN THE SOURCE COORDINATE SYSTEM IN RCS COMPONENTS |
| XTH<br>YTH<br>ZTH | X,Y,Z COMPONENTS OF THE THETA UNIT POLARIZATION VECTOR OF THE FIELD IN THE SOURCE COORDINATE SYSTEM IN RCS COMPONENTS |

## CODE LISTING

```
C-----------------------------------------------------------------------
      SUBROUTINE SOURCP(EIPRP,EIPLP,VI,PHO,BOP,VAX)
C!!!
C!!!  INCIDENT SLOPE FIELD
C!!!
      COMPLEX EIPRP,EIPLP
      DIMENSION VI(3),PHO(3),BOP(3)
      DIMENSION VAX(3,3)
      LOGICAL LSOR
      COMMON/FARP/IM,H,HAW
      COMMON/PIS/PI,TPI,DPR,RPD
      COMMON/SOURSF/FACTOR
C!!!  TAKE DOT PRODUCTS OF VI AND PRIMED AXES TO OBTAIN THE SINE
C!!!  AND COSINE OF THP AND PHP
      RDX=VI(1)*VAX(1,1)+VI(2)*VAX(1,2)+VI(3)*VAX(1,3)
      RDY=VI(1)*VAX(2,1)+VI(2)*VAX(2,2)+VI(3)*VAX(2,3)
      CTHP=VI(1)*VAX(3,1)+VI(2)*VAX(3,2)+VI(3)*VAX(3,3)
      STHP=SQRT(RDX*RDX+RDY*RDY)
      CPHP=RDX/STHP
      SPHP=RDY/STHP
C!!!  CALCUALTE THETA AND PHI POL. UNIT VECTORS FOR RAY
C!!!  IN SOURCE COORD SYS (IN RCS COMPONENTS)
      XTH=VAX(1,1)*CPHP*CTHP+VAX(2,1)*SPHP*CTHP-VAX(3,1)*STHP
      YTH=VAX(1,2)*CPHP*CTHP+VAX(2,2)*SPHP*CTHP-VAX(3,2)*STHP
      ZTH=VAX(1,3)*CPHP*CTHP+VAX(2,3)*SPHP*CTHP-VAX(3,3)*STHP
      XPH=-SPHP*VAX(1,1)+CPHP*VAX(2,1)
      YPH=-SPHP*VAX(1,2)+CPHP*VAX(2,2)
      ZPH=-SPHP*VAX(1,3)+CPHP*VAX(2,3)
C!!!  CALCULATE SLOPE  INCIDENT FIELDS
      EA=COS(PI*H*CTHP)
      EB=PI*H*STHP*STHP*SIN(PI*H*CTHP)
      ACTHP=ABS(CTHP)
      IF(ABS(ACTHP-.5/H).LT.1.E-5) GO TO 5
      E1=1.-4.*H*H*CTHP*CTHP
      E2=1.-4.*H*H*(2.-CTHP*CTHP)
      EFA=(E2*EA*CTHP/E1+EB)/E1
      EFB=EA/E1
      GO TO 6
    5 SN=SIGN(1.,CTHP)
      EFA=SN*(PI+4.*H*H*PI)/16./H
      EFB=PI/4.
    6 CONTINUE
C!!!  COMPUTE DOT PRODUCTS OF RAY POLARIZATION UNIT VECTORS
C!!!  AND UNIT VECTORS PARALLEL AND PERPENDICULAR TO EDGE
      TPHO=XTH*PHO(1)+YTH*PHO(2)+ZTH*PHO(3)
      TPBO=XTH*BOP(1)+YTH*BOP(2)+ZTH*BOP(3)
      PPHO=XPH*PHO(1)+YPH*PHO(2)+ZPH*PHO(3)
      PPBO=XPH*BOP(1)+YPH*BOP(2)+ZPH*BOP(3)
      EFD=0.
      EFC=0.
      EFE=1.0
      IF(HAW.LT.0.1) GOTO 8
      ARG=PI*HAW*STHP*CPHP
      IF(ABS(ARG).LT.1.E-05) ARG=1.0E-05
      SNARG=SIN(ARG)/ARG
      EFC=HAW*(COS(ARG)-SNARG)
      IF(ABS(CPHP).LT.1E-05) CPHP=1E-05
      EFD=SPHP/CPHP*HAW*(SNARG-COS(ARG))
      EFE=HAW*SNARG
    8 EFF=EFA*EFE*CTHP+EF[illegible]*EFC
      IF(IM.EQ.1) GO TO 10
      EX1=EFF*TPHO*[illegible]*PPHO*TPHO*CTHP*EFE*EFB*PPHO*PPBO
      EX2=EFF*TPHO*TPBO[illegible]*PPHO*TPBO*CTHP*EFE*EFB*PPHO*PPBO
      EIPRP=-2.*H*CMPLX(0.,EX1)*6[illegible].*FACTOR
      EIPLP=-2.*H*CMPLX(0.,EX2)*60.*FACTOR
      RETURN
   10 CONTINUE
      EX1=EFF*TPHO*PPHO+EFB*EFD*PPHO*PPHO+CTHP*EFE*EFB*PPHO*TPHO
      EX2=EFF*TPHO*PPBO+EFB*EFD*PPHO*PPBO-CTHP*EFE*EFB*PPHO*TPBO
      EIPRP=2.*H*CMPLX(0.,EX1)*FACTOR/TPI
      EIPLP=2.*H*CMPLX(0.,EX2)*FACTOR/TPI
      RETURN
      END
```

## TANG

PURPOSE

To compute vectors from a source that are tangent to the cylinder in the x-y plane.

PERTINENT GEOMETRY

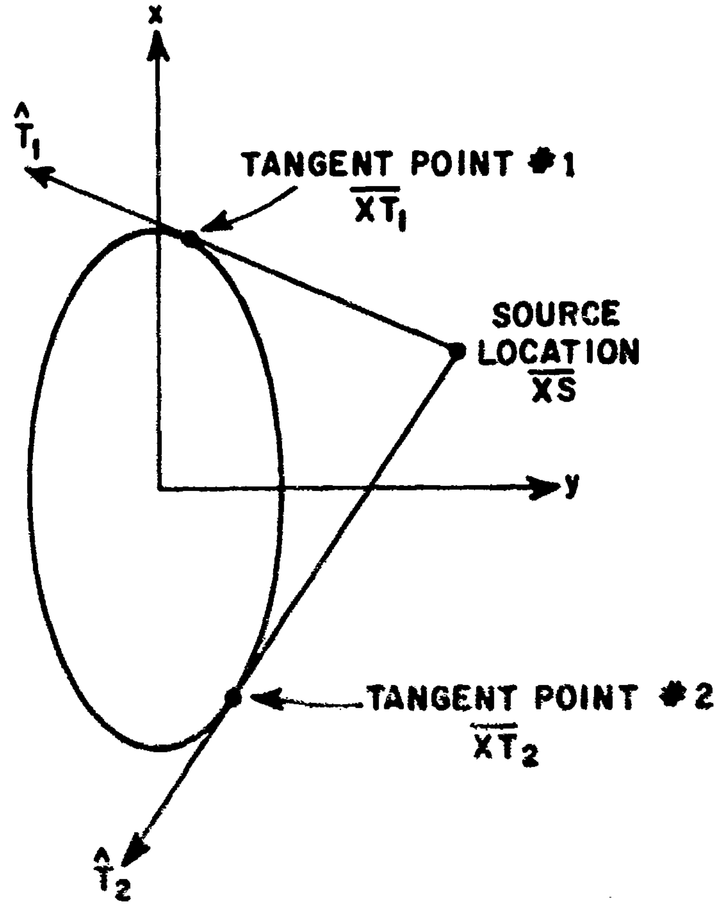


Figure 110--Geometry of source vectors tangent to the cylinder in the x-y plane.

$$\overline{XT}_1 = \hat{x}\ A \cos(VT(1)) + \hat{y}\ B \sin(VT(1))$$

$$\overline{XT}_2 = \hat{x}\ A \cos(VT(2)) + \hat{y}\ B \sin(VT(2))$$

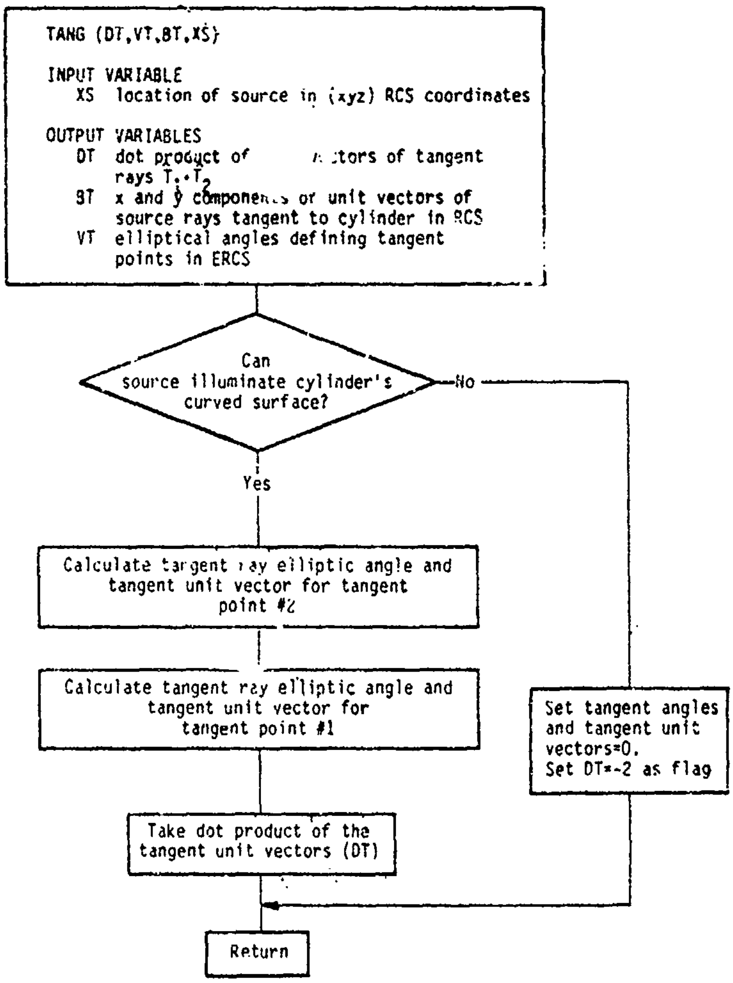
$$\hat{T}_1 = \hat{x}\ BT(1) + \hat{y}\ BT(2)$$

$$\hat{T}_2 = \hat{x}\ BT(3) + \hat{y}\ BT(4)$$

METHOD

The unit tangent vectors are determined by solving a set of equations found by setting the incident vector from the source equal to the general unit tangent vector to the elliptic surface. Details are given in pages 90-93 in Reference 1.

FLOW DIAGRAM


TANG (DT,VT,BT,XS)
INPUT VARIABLE
XS location of source in (xyz) RCS coordinates
OUTPUT VARIABLES
DT dot product of [illegible] vectors of tangent rays $T_1 \cdot T_2$
BT x and y components of unit vectors of source rays tangent to cylinder in RCS
VT elliptical angles defining tangent points in ERCS
Can source illuminate cylinder's curved surface?
No
Yes
Calculate tangent ray elliptic angle and tangent unit vector for tangent point #2
Calculate tangent ray elliptic angle and tangent unit vector for tangent point #1
Set tangent angles and tangent unit vectors=0. Set DT=-2 as flag
Take dot product of the tangent unit vectors (DT)
Return

## SYMBOL DICTIONARY

| | |
|---|---|
| AA | DISTANCE FROM SOURCE TO TANGENT POINT |
| AL | COMPUTATIONAL VARIABLE |
| BB | DISTANCE FROM ORIGIN TO TANGENT POINT |
| BET | COMPUTATIONAL VARIABLE |
| BT | X AND Y COMPONENTS OF TANGENT UNIT VECTORS IN REF COORD SYS. |
| CV | COSINE OF TANGENT POINT ELL ANGLE |
| CVE | COSINE OF VE |
| DT | DOT PRODUCT OF UNIT VECTORS OF THE TWO SOURCE RAYS TANGENT TO THE CYLINDER (2-D) |
| DV1 | ANGLE V1 IN DEGREES |
| DV2 | ANGLE V2 IN DEGREES |
| E1 | ERROR DETECTION VARIABLE |
| E2 | ERROR DETECTION VARIABLE |
| RHOE | DISTANCE FROM Z AXIS TO POINT WHERE RAY FROM ORIGIN TO SOURCE INTERSECTS THE CYLINDER |
| RHOS | DISTANCE FROM SOURCE TO Z AXIS |
| SV | SINE OF TANGENT POINT ELL ANGLE |
| SVE | SINE OF VE |
| SX | X COMPONENT OF RAY FROM TANGENT POINT TO SOURCE |
| SY | Y COMPONENT OF RAY FROM TANGENT POINT TO SOURCE |
| T1X | X COMPONENT OF TANGENT RAY UNIT VECTOR (TAN POINT #2) |
| T1Y | Y COMPONENT OF TANGENT RAY UNIT VECTOR (TAN POINT #2) |
| T2X | X COMPONENT OF TANGENT RAY UNIT VECTOR (TAN POINT #1) |
| T2Y | Y COMPONENT OF TANGENT RAY UNIT VECTOR (TAN POINT #1) |
| V1 | ELL ANGLE DEFINING TANGENT POINT #2 |
| V2 | ELL ANGLE DEFINING TANGENT POINT #1 |
| VE | ELL ANGLE OF RAY FROM ORIGIN TO SOURCE |
| VT | ELL ANGLE DEFINING TANGENT POINT LOCATION IN ERCS |
| XS | SOURCE LOCATION |
| XT | X-COMPONENT OF TANGENT POINT LOCATION |
| YT | Y-COMPONENT OF TANGENT POINT LOCATION |
| XY | COMPUTATIONAL VARIABLE |

## CODE LISTING

```
C------------------------------------------------------------------
      SUBROUTINE TANG(DT,VT,BT,XS)
C!!!
C!!!  COMPUTES TANGENT VECTORS TO ELLIPTIC CYLINDER FROM SOURCE
C!!!
      DIMENSION VT(2),BT(4),XS(3)
      COMMON/PIS/PI,TPI,DPR,RPD
      COMMON/GEOMEL/A,B,ZC(2),SNC(2),CNC(2),CTC(2)
      RHOS=SQRT(XS(1)*XS(1)+XS(2)*XS(2))
C!!!  CAN SOURCE ILLUMINATE CYLINDER SURFACE?
      IF(RHOS.GT.A.AND.RHOS.GT.B) GO TO 20
      IF(RHOS.LT.A.AND.RHOS.LT.B) GO TO 10
      VE=BTAN2(A*XS(2),B*XS(1))
      CVE=COS(VE)
      SVE=SIN(VE)
      RHOE=SQRT(A*A*CVE*CVE+B*B*SVE*SVE)
C!!!  IF SOURCE CANNOT ILLUMINATE CYLINDER, SET ANGLES
C!!!  TO ZERO, SET FLAG, AND RETURN
      IF(RHOS.GE.RHOE) GO TO 20
 10   CONTINUE
      DT=-2.
      VT(1)=0.
      VT(2)=0.
      BT(1)=0.
      BT(2)=0.
      BT(3)=0.
      BT(4)=0.
      RETURN
 20   CONTINUE
      XY=B*B*XS(1)*XS(1)+A*A*XS(2)*XS(2)
      AL=A*A*B*B/XY
      BET=SQRT(XY-A*A*B*B)/XY
C!!!  CALCULATE TAN ANGLE AND TAN UNIT VECTOR FOR TAN POINT #2
      XT=AL*XS(1)+A*A*BET*XS(2)
      YT=AL*XS(2)-B*B*BET*XS(1)
      V1=BTAN2(A*YT,B*XT)
      SV=SIN(V1)
      CV=COS(V1)
      SX=XS(1)-A*CV
      SY=XS(2)-B*SV
      AA=SQRT(SX*SX+SY*SY)
      BB=SQRT(A*A*SV*SV+B*B*CV*CV)
      E1=SQRT((SX/AA+A*SV/BB)**2+(SY/AA-B*CV/BB)**2)
      T1X=A*SV/BB
      T1Y=-B*CV/BB
C!!!  CALCULATE TAN ANGLE AND TAN UNIT VECTOR FOR TAN POINT #1
      XT=AL*XS(1)-A*A*BET*XS(2)
      YT=AL*XS(2)+B*B*BET*XS(1)
      V2=BTAN2(A*YT,B*XT)
      SV=SIN(V2)
      CV=COS(V2)
      SX=XS(1)-A*CV
      SY=XS(2)-B*SV
      AA=SQRT(SX*SX+SY*SY)
      BB=SQRT(A*A*SV*SV+B*B*CV*CV)
      E2=SQRT((SX/AA-A*SV/BB)**2+(SY/AA+B*CV/BB)**2)
      T2X=-A*SV/BB
      T2Y=B*CV/BB
C!!!  TAKE DOT PRODUCT OF TANGENT UNIT VECTORS
      DT=T1X*T2X+T1Y*T2Y
      DV1=V1*DPR
      DV2=V2*DPR
      VT(1)=V2
      VT(2)=V1
      BT(1)=T2X
      BT(2)=T2Y
      BT(3)=T1X
      BT(4)=T1Y
      IF(E1.GT.1.E-5)WRITE(6,1)DV1,E1
      IF(E2.GT.1.E-5)WRITE(6,1)DV2,E2
    1 FORMAT(1H ,'ERROR IN TANGENT SECTION: ',2F1Ø.5)
      RETURN
      END
```

# CHAPTER V
# COMMON BLOCK

This chapter defines the variables used in common blocks. The blocks are arranged in alphabetical order.

COMMON BNDDCL--------------------

THIS COMMON BLOCK CONTAINS INFORMATION CONCERNING THE STARTING POINT PARAMETERS AND BOUNDS FOR TRACING A RAY DIFFRACTED FROM A PLATE EDGE AND THEN REFLECTED FROM THE CYLINDER. THE INFORMATION IS GENERATED IN SUBROUTINE GEOMPC AND IS USED IN SUBROUTINE DPLRCL.

VDC(14,6) THIS ARRAY CONTAINS THE ELLIPTIC ANGLE VDC(MP,ME) DEFINING THE STARTING REFLECTION POINT ON THE CYLINDER FOR A RAY DIFFRACTED FROM EDGE ME OF PLATE MP AND THEN REFLECTED BY THE CYLINDER

UDC(2) THIS ARRAY CONTAINS THE LINEAR VALUE UDC(N) DEFINING THE Z COMPONENT OF THE STARTING REFLECTION POINTS ON THE CYLINDER AXIS. UDC(1) IS FOR THE MOST POSITIVE Z LOCATION AND UD(2) IS FOR THE MOST NEGATIVE Z LOCATION.

PDCR(14,6,2) THIS ARRAY CONTAINS ANGLES PDCR(MP,ME,N) DEFINING THE PHI COMPONENT OF THE REFL RAY DIRECTION OF RAYS DIF BY EDGE ME OF PLATE MP AND THEN REFLECTED AT STARTING POINT N ON THE CYLINDER

TDCR(14,6,2) THIS ARRAY CONTAINS ANGLES TDCR(MP,ME,N) DEFINING THE REFL RAY THETA COMPONENT OF RAY DIRECTIONS FOR RAYS DIF BY EDGE ME OF PLATE MP AND THEN REFLECTED BY STARTING REFLECTION POINT N ON THE CYLINDER.

DTDC(14,6) DOT PRODUCT OF UNIT VECTORS OF RAYS DIFFRACTED BY EDGE ME OF PLATE MP AND REFLECTED BY THE PREFERRED STARTING POINT ON THE CYLINDER

BTDC(14,6,4) THIS ARRAY CONTAINS VARIABLES DEFINING THE VECTORS HAVING BEEN DIFFRACTED BY THE CORNER OF EDGE ME OF PLATE MP FURTHEST FROM THE CYLINDER WHICH ARE TANGENT TO THE CYLINDER.
THE TWO TANGENT VECTORS ARE GIVEN BY:

$\widehat{T1}=\hat{X}*BTDC(MP,ME,1)+\hat{Y}*BTDC(MP,ME,2)$

$\widehat{T2}=\hat{X}*BTDC(MP,ME,3)+\hat{Y}*BTDC(MP,ME,4)$

DDC(14,6,2) THIS ARRAY CONTAINS THE COSINE OF THE STARTING REFLECTED RAY THETA ANGLE, WHERE

DDC(MP,ME,N)=COS(TDCR(MP,ME,N))

COMMON BNDFCL --------------------

THIS COMMON BLOCK IS GENERATED IN SUBROUTINE GEOM AND IS USED TO SPECIFY THE PERMISSABLE RANGE OF DIFFRACTION ANGLES FOR SOURCE RAYS DIFFRACTED BY A PLATE EDGE.

BD(14,6,2) THIS DEFINES PERMISSABLE THETA DIFFRACTION ANGLES FOR WEDGE DIFFRACTION
THE PERMISSABLE RANGE FOR DIFFRACTION ANGLE B FOR A SOURCE RAY DIFFRACTED BY EDGE ME OF PLATE MP IS GIVEN BY:

COS(B1)<COS(BØ)<COS(B2)

WHERE BO IS THE ANGLE THE DIFFRACTED RAY MAKES WITH THE EDGE, AND B1 AND B2 ARE DEFINED AT THE CORNERS OF THE PLATE AS

COS(B1)=BD(MP,ME,1)
COS(B2)=BD(MP,ME,2).

COMMON BNDICL --------------------

THIS COMMON BLOCK CONTAINS INFORMATION RELATED TO VECTORS REFLECTED FROM PLATES WHICH ARE TANGENT TO THE CYLINDER. THE DATA IS GENERATED IN GEOMPC.

DTI(14) THIS IS THE DOT PRODUCT OF THE TWO RAYS REFLECTED BY PLATE MP WHICH ARE TANGENT TO THE CYLINDER THE CYLINDER FROM THE SOURCE IMAGE FOR REFLECTION FROM PLATE MP:

$DTI(MP)=\widehat{T1}\cdot\widehat{T2}$

VTI(14,2) THIS IS AN ARRAY OF ELLIPTICAL ANGLES DEFINING THE TWO TANGENT POINTS ON THE CYL FOR RAYS WHICH ARE REFLECTED FROM PLATE MP AND TANGENT TO THE CYLINDER. TANGENT POINT N FOR RAY REFLECTED FROM PLATE MP ARE GIVEN BY:

```
                X=A*COS(VTI(MP,N))
                Y=B*SIN(VTI(MP,N))
BTI(14,4)   THIS DEFINES UNIT VECTORS OF THE TWO RAYS REFLECTED
          BY PLATE MP AND TANGENT TO THE CYLINDER.
          THE UNIT VECTOR FOR THE SOURCE RAY REFLECTED FROM
          PLATE MP TANGENT TO TAN POINT 1 IS
          GIVEN BY:
                T1=X*BTI(MP,1)+Y*BTI(MP,2)
          THE UNIT VECTOR FOR THE SOURCE RAY REFLECTED FROM
          PLATE MP TANGENT TO TAN POINT 2
          IS GIVEN BY:
                T2=X*BTI(MP,3)+Y*BTI(MP,4)

COMMON BNDRCL  -------------------
THIS COMMON BLOCK CONTAINS INFORMATION CONCERNING THE
STARTING PARAMETERS AND BOUNDS FOR RAYS REFLECTED FROM THE
CYLINDER AND THEN DIFFRACTED FROM A PLATE EDGE.  THE INFORMATION
IS GENERATED IN SUBROUTINE GEOMPC AND IS USED IN SUBROUTINE RCLDPL.
VCD(14,6)  THIS ARRAY CONTAINS THE ELLIPTIC ANGLE VCD(MP,MC)
          THAT DEFINES THE X,Y COMPONENTS OF THE REFLECTION
          POINT LOCATION FOR THE RAY WHICH IS REFLECTED BY
          THE CYLINDER AND HITS CORNER MC OF PLATE MP.
UCD(14,6)  THIS ARRAY CONTAINS THE LINEAR VALUE UCD(MP,MC)
          THAT DEFINES THE Z COMPONENT OF THE REFLECTION POINT
          FOR THE RAY THAT IS REFLECTED BY THE CYLINDER AND
          HITS CORNER MC OF PLATE MP.
          THE REFLECTION POINT LOCATION IS GIVEN BY
                X=A*COS(VCD(MP,ME))
                Y=B*COS(VCD(MP,MC))
                Z=UDC(MP,MC)
BCD(14,6,2)  THIS ARRAY CONTAINS THE VALUE BCD(MP,ME,N)
          THAT DEFINES THE PERMISSABLE RANGE OF THE BETA
          DIFFRACTION ANGLES FOR THE RAY THAT IS REFL BY THE
          CYLINDER AND DIFFRACTED BY EDGE ME OF PLATE MP.
          THE PERMISSABLE RANGE FOR DIFFRACTION ANGLE BO FOR
          THIS RAY IS GIVEN BY:
                COS(B1)<COS(BO)<COS(B2)
          WHERE BO IS THE ANGLE THE DIFFRACTED RAY MAKES WITH
          THE EDGE AND ANGLES B1 AND B2 ARE DEFINED AT THE
          CORNERS OF THE PLATE AS:
                COS(B1)=BCD(MP,ME,1)
                COS(B2)=BCD(MP,ME,2)

COMMON BNDSCL -----------------
THIS COMMON BLOCK CONTAINS INFORMATION RELATED TO VECTORS
FROM THE SOURCE THAT ARE TANGENT TO THE CYLINDER.
THE DATA IS GENERATED IN SUBROUTINE GEOMC
DTS       THIS IS THE DOT PRODUCT OF THE TWO SOURCE VECTORS
          TANGENT TO THE CYLINDER:
                DTS=T1 T2
VTS(2)  VTS CONSISTS OF TWO ELLIPTICAL ANGLES DEFINING THE
          TWO TANGENT POINTS ON THE CYLINDER.
          TANGENT POINT N IS GIVEN BY:
                X=A*COS(VTS(N))
                Y=B*SIN(VTS(N))
BTS(4)  THIS DEFINES UNIT VECTORS OF THE TWO SOURCE
          RAYS TANGENT TO THE CYLINDER.
          THE UNIT VECTOR FOR THE SOURCE RAY TANGENT TO
          TAN POINT 1 IS GIVEN BY:
                T1=X*BTS(1)+Y*BTS(2)
          THE UNIT VECTOR FOR THE SOURCE RAY TANGENT TO
          TAN POINT 2 IS GIVEN BY:
                T2=X*BTS(3)+Y*BTS(4)

COMMON BRDPHN  ------------------------
THIS COMMON BLOCK IS GENERATED IN SUBROUTINE GEOMPC AND IS
```

USED TO SPECIFY THE BRANCH CUT DISPLACEMENT ANGLE FOR THE PLATE-CYLINDER REFLECTED-DIFFRACTED AND DIFFRACTED-REFLECTED TERMS.

PHWR(14,6) IS THE PHI ANGLE LOCATION OF THE CENTER OF EDGE ME OF PLATE MP, WITH RESPECT TO THE CYLINDER

COMMON CLDRC-------------------------

THIS COMMON BLOCK CONTAINS AN ARRAY OF VARIABLES WHICH ARE GENERATED IN MAIN AND SUBROUTINE DPLRCL AND ARE PASSED THROUGH A SUBROUTINE WINDOW TO SUBROUTINE DFRFPT WHERE THEY ARE USED

LDRC(14,6) IS AN ARRAY OF LOGICAL VARIABLES.
LDRC(MP,ME) IS SET TRUE IF STARTING POINT DATA IS AVAILABLE FROM PREVIOUS PATTERN ANGLE (FOR NEXT PATTERN ANGLE) WHEN DEFINING THE REFLECTION POINT ON CYLINDER FOR A RAY WHICH IS DIFFRACTED FROM EDGE ME OF PLATE MP AND THEN REFLECTED BY THE CYLINDER

COMMON CLRDC -------------------------

THIS COMMON BLOCK CONTAINS AN ARRAY OF VARIABLES WHICH ARE GENERATED IN MAIN AND SUBROUTINE RCLDPL AND ARE PASSED THROUGH A SUBROUTINE WINDOW TO SUBROUTINE RFDFPT, WHERE THEY ARE USED

LRDC(14,6) IS AN ARRAY OF LOGICAL VARIABLES.
LRDC(MP,ME) IS SET TRUE IF STARTING POINT DATA IS AVAILABLE FROM PREVIOUS PATTERN ANGLE (FOR NEXT PATTERN ANGLE) WHEN DEFINING THE REFLECTION POINT ON CYLINDER FOR A RAY WHICH IS REFLECTED BY THE CYLINDER AND THEN DIFFRACTED BY EDGE ME OF PLATE MP

COMMON CLRFC ------------------

THIS COMMON BLOCK CONTAINS ONE VARIABLE WHICH IS GENERATED IN MAIN AND SUBROUTINE REFCYL AND IS PASSED THROUGH A SUBROUTINE WINDOW TO SUBROUTINE RFPTCL, WHERE IT IS USED

LRFC IS A LOGICAL VARIABLE WHICH IS SET TRUE IF THE STARTING POINT DATA IS AVAILABLE FROM PREVIOUS PATTERN ANGLE (FOR NEXT PATTERN ANGLE) WHEN DEFINING THE REFLECTION POINT ON THE CYLINDER

COMMON CLRFI ------------------

THIS COMMON BLOCK CONTAINS AN ARRAY OF VARIABLES WHICH ARE GENERATED IN MAIN AND SUBROUTINE RPLRCL AND ARE PASSED THROUGH A SUBROUTINE WINDOW TO SUBROUTINE RFPTCL, WHERE THEY ARE USED

LRFI(14) IS AN ARRAY OF LOGICAL VARIABLES. LRFI(MP) IS SET TRUE IF STARTING POINT DATA IS AVAILABLE FROM PREVIOUS PATTERN ANGLE (FOR NEXT PATTERN ANGLE) WHEN DEFINING REFLECTION POINT ON THE CYLINDER FOR A RAY REFLECTED BY PLATE MR AND THEN REFLECTED BY THE CYLINDER

COMMON CLRFS --------------------

THIS COMMON BLOCK CONTAINS AN ARRAY OF VARIABLES WHICH IS GENERATED IN MAIN AND SUBROUTINE RCLRPL AND IS PASSED THROUGH A SUBROUTINE WINDOW TO SUBROUTINE RFPTCL, WHERE IT IS USED.

LRFS(14) IS AN ARRAY OF LOGICAL VARIABLES.
LRFS(MP) IS SET TRUE IF STARTING POINT DATA IS AVAILABLE FOR THE NEXT PATTERN ANGLE WHEN DEFINING THE REFLECTION POINT ON A CYLINDER FOR A RAY REFLECTED BY THE CYLINDER AND THEN REFLECTED BY PLATE MP.

COMMON COMP -------------------

THIS COMMON BLOCK CONTAINS TWO CONSTANTS USED THROUGHOUT THE PROGRAM

CJ THE IMAGINARY CONSTANT, J (=SQRT(-1))
CPI4 THE COMPLEX CONSTANT, CEXP(-J*PI/4)

```
COMMON DIR  --------------------
THIS COMMON BLOCK CONTAINS INFORMATION SPECIFYING THE DIRECTION OF
PROPAGATION (THE DESIRED OBSERVATION DIRECTION).
THE INFORMATION IS COMPUTED IN THE MAIN PROGRAM
D(3)    THE UNIT VECTOR OF THE PROPAGATION DIRECTION IN
        (XYZ) REFERENCE COORDINATE SYSTEM COMPONENTS:
              D̂=X̂*D(1)+Ŷ*D(2)+Ẑ*D(3)
THSR    THETA ANGLE DEFINING PROPAGATION DIRECTION IN SPHERICAL REFERENCE
        COORDINATE SYSTEM (MEASURED FROM Z-AXIS) IN RADIANS
PHSR    PHI ANGLE DEFINING PROPAGATION DIRECTION IN SPHERICAL REFERENCE
        COORDINATE SYSTEM (MEASURED FROM X-AXIS) IN RADIANS
SPS     THE SINE OF PHSR
CPS     THE COSINE OF PHSR
STHS    THE SINE OF THSR
CTHS    THE COSINE OF THSR

COMMON DOUBLE  ----------------
THIS COMMON BLOCK CONTAINS INFORMATION DEFINING ANGLES WHERE
DOUBLE DIFFRACTION TERMS WOULD BE SIGNIFICANT (SHADOW
BOUNDARIES FOR SINGLE DIFFRACTED RAYS)
IDD(361)  THIS INTEGER IDENTIFIES WHICH EDGE THE FIRST
        DIFFRACTION OCCURS FROM AND WHICH PLATE SHADOWS IT
        FOR A GIVEN PATTERN ANGLE, II
ID(14,6)  THIS INTEGER ARRAY IS USED TO STORE THE PLATE THAT
        SHADOWS THE RAY DIFFRACTED FROM EDGE ME OF PLATE
        MP (ID(ME,MP)).
II      THIS INTEGER VARIABLE IDENTIFIES THE OBSERVATION
        ANGLE UNDER CONSIDERATION

COMMON EDMAG  -----------------------
THIS COMMON BLOCK IS GENERATED IN SUBROUTINE GEOM AND IS USED TO
DEFINE PLATE EDGE LENGTHS
VMAG(14,6)  THIS DEFINES THE LENGTH OF EDGES ON PLATES IN WAVELENGTHS.
        THE LENGTH OF EDGE ME OF PLATE MP IS GIVEN BY
             VMAG(MP,ME)

COMMON ESTOR  -----------------
THIS COMMON BLOCK IS USED IN MAIN TO STORE THE TOTAL ELECTRIC
FIELDS.
ETHT(361)  THIS COMPLEX ARRAY IS USED TO STORE THE TOTAL
        E-THETA FIELD
EPHT(361)  THIS COMPLEX ARRAY IS USED TO STORE THE TOTAL
        E-PHI FIELD

COMMON FARP  ------------------
THIS COMMON BLOCK DEFINES THE TYPE OF SOURCE USED AND THE
DIMENSIONS OF THE SOURCE (VARIABLES DEFINED IN MAIN PROGRAM)
IM      THIS DEFINES THE TYPE OF SOURCE USED:
               IM=0 SPECIFIES ELECTRIC SOURCE
               IM=1 SPECIFIES MAGNETIC SOURCE
H       THE LENGTH OF THE SOURCE (IN THE DIRECTION OF THE SOURCE
        CURRENT) IN WAVELENGTHS
HAW     THE APERTURE WIDTH IN WAVELENGTHS (WIDTH OF THE SOURCE)
        (IF HAW IS LESS THAN 0.1 WAVELENGTHS , THE CODE
        ASSUMES THE SOURCE TO BE A LINE SOURCE)

COMMON FEDPAT  ---------------------
THIS COMMON BLOCK CONTAINS SOURCE PATTERN
FACTOR INFORMATION FOR USE WHEN THE USER
CHOOSES TO DEFINE THE SOURCE PATTERN FROM DATA OBTAINED ELSEWHERE
TO BE USED IN AN INTERPOLATION SCHEME
EFED(361)  THIS COMPLEX ARRAY DEFINES THE E-PLANE PATTERN OF THE
        SOURCE
HFED(361)  THIS COMPLEX ARRAY DEFINES THE H-PLANE PATTERN OF THE
        SOURCE
```

```
COMMON FNANG -------------------
THIS COMMON BLOCK DEFINES WEDGE ANGLES FOR PLATE EDGES.  IT IS
GENERATED IN SUBROUTINE GEOM AND USED IN DIFFRACTION COEFFICIENT
CALCULATIONS.
FNP(14,6)  WEDGE ANGLE OF EDGE ME OF PLATE MP
       FNP(MP,ME)=(2*PI-WA)/PI ,WHERE WA IS THE INSIDE
       ANGLE OF THE WEDGE.  IT IS RENAMED FN IN THE MAIN
       PROGRAM BEFORE CALLING DIFFRACTION SUBROUTINES
       NOTE: IF TWO PLATES INTERSECT, DIFFRACTION CALCULATION IS ONLY
       CALCULATED ONCE, EVEN THOUGH TWO DIFFERENT EDGES ARE INVOLVED.
       THEREFORE, THE WEDGE ANGLE FOR ONE OF THE COMMON EDGES
       WILL BE SET NEGATIVE AS A FLAG AND THE DIFFRACTED FIELD
       WILL ONLY BE CALCUALTED ONCE FOR THE COMMON EDGES
       (THE FLAGGED EDGE IS IGNORED)

COMMON FUDG  ---------------------------
THIS COMMON BLOCK IS USED TO TRANSFER DATA CONCERNING GEOMETRICAL
OPTICS REFLECTION FROM THE CYLINDER IN SUBROUTINE REFCYL TO
SUBROUTINE SCTCYL
TRAN    THE SPREAD FACTOR AND PHASE OF THE G.O. FIELD
ESTH }  THETA AND PHI COMPONENTS OF SOFT COMPONENT OF FIELD INCIDENT
ESPH }  ON CYLINDER REFLECTION POINT
EHTH }  THETA AND PHI COMPONENTS OF HARD COMPONENT OF FIELD INCIDENT
EHPH }  ON CYLINDER REFLECTION POINT
XR(3)   X,Y,Z COMPONENTS OF THE REFLECTION POINT LOCATION IN RCS
RG      RADIUS OF CURVATURE OF CYLINDER AT REFLECTION POINT
RHOI    RAY SPREADING RADIUS IN PLANE OF CYLINDER CURVATURE AT
        REFLECTION POINT IN RCS
SMAG    DISTANCE FROM SOURCE TO REFLECTION POINT
LTRF    SET TRUE IF GEOMETRICAL OPTICS REFLECTED FIELD
        IS NOT PRESENT

COMMON FUDGI  ------------------
THIS COMMON BLOCK IS USED TO TRANSFER DATA CONCERNING
GEOMETRICAL OPTICS REFLECTION FROM A PLATE THEN FROM THE
CYLINDER IN SUBROUTINE RPLRCL TO SUBROUTINE RPLSCL.
TRAN    THE SPREAD FACTOR AND PHASE OF THE GEOMETRICAL OPTICS
        FIELD
ESTH    THE THETA COMPONENT OF THE SOFT COMPONENT OF THE
        FIELD INCIDENT ON CYLINDER REFLECTION POINT AFTER
        PLATE REFLECTION
ESPH    PHI COMPONENT OF SOFT COMPONENT OF THE FIELD INCIDENT
        ON THE CYLINDER REFLECTION POINT AFTER PLATE REFL.
EHTH    THETA COMPONENT OF HARD COMPONENT OF FIELD
        INCIDENT ON CYLINDER REFLECTION POINT AFTER PLATE REFLECTION
EHPH    PHI COMPONENT OF HARD COMPONENT OF FIELD INCIDENT
        ON CYLINDER REFLECTION POINT AFTER PLATE REFLECTION
XR(3)   X,Y,Z COMPONENTS OF THE REFLECTION POINT LOCATION
        IN RCS
RG      RAY SPREADING RADIUS IN PLANE OF CYLINDER CURVATURE
        AT REFLECTION POINT IN RCS
RHOI    RAY SPREADING RADIUS IN PLANE OF CYLINDER CURVATURE
        AT REFLECTION POINT IN RCS
SMAG    DISTANCE FROM THE SOURCE IMAGE TO THE CYLINDER
        REFLECTION POINT
LTRFI   SET TRUE IF GEOMETRICAL OPTICS REFLECTED FIELD
        IS NOT PRESENT.

COMMON FUDGJ --------------------------
THIS COMMON BLOCK IS USED TO TRANSFER DATA CONCERNING GEOMETRICAL
OPTICS REFLECTION FROM THE CYLINDER AND THEN A PLATE IN
SUBROUTINE RCLRPL TO SUBROUTINE SCLRPL
TRAN    THE SPREAD FACTOR AND PHASE OF THE G.O. FIELD
ESTH }  THETA AND PHI COMPONENTS OF SOFT COMPONENT OF FIELD INCIDENT
ESPH }  ON CYLINDER REFLECTION POINT
```

```
EHTH    THETA AND PHI COMPONENTS OF HARD COMPONENT OF FIELD INCIDENT
EHPH    ON CYLINDER REFLECTION POINT
XR(3)   X,Y,Z COMPONENTS OF THE REFLECTION POINT LOCATION IN RCS
RG      RADIUS OF CURVATURE OF CYLINDER AT REFLECTION POINT
RHC1    RAY SPREADING RADIUS IN PLANE OF CYLINDER CURVATURE AT
        REFLECTION POINT IN RCS
SMAG    DISTANCE FROM SOURCE TO REFLECTION POINT
LTRFJ   SET TRUE IF GEOMETRICAL OPTICS REFLECTED FIELD
        IS NOT PRESENT

COMMON GEOMEL  -------------------
THIS COMMON BLOCK CONTAINS INFORMATION DEFINING THE ELLIPTIC
CYLINDER GEOMETRY (SPECIFIED IN MAIN PROGRAM FROM DATA INPUT)
A       RADIUS OF ELL CYLINDER ALONG X-AXIS OF
        THE CYLINDER IN WAVELENGTHS
B       RADIUS OF ELL CYLINDER ALONG Y-AXIS OF
        THE CYLINDER IN WAVELENGTHS
ZC(2)   POINT WHERE END CAP MC INTERSECTS Z AXIS OF REFERENCE
        COORDINATE SYSTEM
        THE VARIABLE ZC(1) REFERS TO THE MOST POSITIVE
        END CAP AND THE ZC(2) REFERS TO THE MOST NEGATIVE
        END CAP
SNC(2)  THIS IS THE SINE OF THE ANGLE BETWEEN THE Z AXIS AND THE
        PLANE OF END CAP MC (ANGLE MEASURED IN X-Z PLANE)
CNC(2)  THIS IS THE COSINE OF THE ANGLE BETWEEN THE Z AXIS AND THE
        PLANE OF END CAP MC (ANGLE MEASURED IN X-Z PLANE)
CTC(2)  THIS IS THE COTANGENT OF THE ANGLE BETWEEN THE Z AXIS AND THE
        PLANE OF END CAP MC (ANGLE MEASURED IN X-Z PLANE)

COMMON GEOPLA  ----------------
THIS COMMON BLOCK CONTAINS GEOMETRICAL DATA DEFINING THE
GEOMETRY OF THE PLATES (CALCULATED IN SUBROUTINE GEOM)
X(14,6,3)  THIS ARRAY DEFINES CORNER LOCATIONS FOR ALL OF THE PLATES IN
        THE (XYZ) REFERENCE COORDINATE SYSTEM COMPONENTS
        IN WAVELENGTHS
        THE LOCATION OF CORNER MC ON PLATE MP IS AS FOLLOWS:
               X=X(MP,MC,1)
               Y=X(MP,MC,2)
               Z=X(MP,MC,3)
V(14,6,3)  THIS DEFINES THE EDGE UNIT VECTOR FOR EACH EDGE ON
        EACH PLATE
        THE EDGE VECTOR V OF EDGE ME ON PLATE MP IS AS FOLLOWS:
               V̂=X̂*V(MP,ME,1)+Ŷ*V(MP,ME,2)+Ẑ*V(MP,ME,3)
               (NOTE THAT EDGE ME IS BETWEEN CORNERS MC AND MC+1
               WHERE MC=ME)
VP(14,6,3)  THIS DEFINES THE UNIT BINORMAL FOR EACH EDGE ON EACH
        PLATE IN (XYZ) REFERENCE SYSTEM COMPONENTS
        THE EDGE BINORMAL FOR EDGE ME OF PLATE MP IS AS FOLLOWS:
               VP̂=X̂*VP(MP,ME,1)+Ŷ*VP(MP,ME,2)+Ẑ*VP(MP,ME,3)
VN(14,3)  THIS DEFINES THE UNIT NORMAL FOR EACH PLATE IN (XYZ)
        REFERENCE COORDINATE SYSTEM COMPONENTS
        THE PLATE UNIT NORMAL FOR PLATE MP IS GIVEN AS FOLLOWS:
               VN̂=X̂*VN(MP,1)+Ŷ*VN(MP,2)+Ẑ*VN(MP,3)
NEP(14)  THIS INTEGER ARRAY DEFINES THE NUMBER OF EDGES
        (OR CORNERS) ON PLATE MP
MPX     THIS INTEGER DEFINES THE NUMBER OF PLATES IN
        THE GEOMETRY (NOT INCLUDING GROUND PLATE)

COMMON GROUND  ------------------
THIS COMMON BLOCK GIVES INFORMATION CONCERNING THE INFINITE GROUND
PLANE
LGRND   A LOGICAL VARIABLE USED TO INDICATE THE PRESENCE OF AN
        INFINITE GROUND PLANE
               LGRND=T  INDICATES GROUND PLANE PRESENT
               LGRND=F  INDICATES GROUND PLANE NOT USED
MPXG    THE MAXIMUM NUMBER OF PLATES PRESENT (INCLUDING THE
        GROUND PLANE IF ONE IS USED)
```

```
COMMON GTD  -----------------
THIS COMMON BLOCK CONTAINS INFORMATION RELATED TO THE
CREEPING WAVES IN SUBROUTINES SCTCYL,RPLSCL,SCL [illegible]
FCT, AND RADCV
AS      PI MINUS THSR (THSR IS THE THETA COMPONENT OF THE
        OBSERVATION DIRECTION IN REFERENCE COORDINATE SYSTEM
        RELATIVE TO THE CYLINDER AXIS IN RADIANS)
IDG     FLAG FOR FUNCTION FCT
SAS     THE SINE OF AS
SASP    THE ABSOLUTE VALUE OF THE SINE OF AS-PI/2
CAS     THE COSINE OF AS

COMMON HITPLT  -----------------
THIS COMMON BLOCK CONTAINS A VARIABLE THAT IS DEFINED
IN SUBROUTINE PLAINT AND IS USED IN SUBROUTINE GEOM
FOR IDENTIFYING DOUBLE DIFFRACTIONS FOR PLATES
MPH     THE NUMBER OF THE PLATE WHICH THE RAY HITS FIRST

COMMON IMAINF  -----------------
THIS COMMON BLOCK DEFINES SOURCE IMAGE LOCATIONS AND
DIRECTIONS FOR REFLECTION FROM PLATES. (CALCULATED IN GEOM)
XI(14,14,3)  THIS GIVES THE SOURCE IMAGE LOCATIONS IN
        WAVELENGTHS FOR ALL SINGLE AND DOUBLE REFLECTIONS
        FROM PLATES
        THE SOURCE IMAGE LOCATION FOR A RAY WHICH IS SINGLY REFLECTED
        FROM PLATE MP IS GIVEN BY:
                X=XI(MP,MP,1)
                Y=XI(MP,MP,2)
                Z=XI(MP,MP,3)
        THE SOURCE IMAGE LOCATION FOR A DOUBLY REFLECTED RAY WHICH
        REFLECTS OFF OF PLATE MP AND THEN PLATE MPP IS GIVEN BY:
                X=XI(MP,MPP,1)
                Y=XI(MP,MPP,2)
                Z=XI(MP,MPP,3)
VXI(3,3,14)  THIS SPECIFIES SINGLE REFLECTION SOURCE IMAGE
        COORDINATE SYSTEM AXES UNIT VECTORS IN (XYZ) REFERENCE
        COORDINATE SYSTEM COMPONENTS
        THE IMAGE SOURCE COORDINATE SYSTEM AXES UNIT VECTORS
        FOR SINGLE REFLECTION OF SOURCE IN PLATE MP ARE
        GIVEN BY:
        GIVEN BY:
                X̂P=X̂*VXI(1,1,MP)+Ŷ*VXI(1,2,MP)+Ẑ*VXI(1,3,MP)
                ŶP=X̂*VXI(2,1,MP)+Ŷ*VXI(2,2,MP)+Ẑ*VXI(2,3,MP)
                ẐP=X̂*VXI(3,1,MP)+Ŷ*VXI(3,2,MP)+Ẑ*VXI(3,3,MP)

COMMON INCINF  -----------------
THIS BLOCK CONTAINS INFORMATION DEFINING THE SOURCE IMAGE
FOR SINGLE REFLECTION FROM A CYLINDER END CAP IN WAVELENGTHS.
THE INFORMATION IS GENERATED IN GEOMC AND IMCDIR.
XIC(2,3)  THIS GIVES THE SOURCE IMAGE LOCATIONS FOR SINGLE
        REFLECTIONS FROM CYLINDER END CAPS.
        THE SOURCE LOCATION FOR REFLECTION FROM
        END CAP MC IS GIVEN IN THE MCS AS:
              X=XIC(MC,1)
              Y=XIC(MC,2)
              Z=XIC(MC,3)
VXIC(3,3,2)  THIS DEFINES THE SOURCE IMAGE COORDINATE
        SYSTEM AXES FOR REFLECTION FROM END CAPS.
        THE SOURCE IMAGE COORDINATE SYSTEM AXES UNIT
        VECTORS FOR A RAY REFLECTED FROM END CAP MC ARE
        GIVEN IN THE MCS AS FOLLOWS:
              X̂P=X̂*VXIC(1,1,MC)+Ŷ*VXIC(1,2,MC)+Ẑ*VXIC(1,3,MC)
              ŶP=X̂*VXIC(2,1,MC)+Ŷ*VXIC(2,2,MC)+Ẑ*VXIC(2,3,MC)
              ẐP=X̂*VXIC(3,1,MC)+Ŷ*VXIC(3,2,MC)+Ẑ*VXIC(3,3,MC)
```

```
COMMON LDCBY  -------------------
THIS ARRAY OF VARIABLES IS COMPUTED IN SUBROUTINE GEOMPC
LDC(14,6)  LOGICAL VARIABLE
       LDC(MP,ME) IS SET TRUE IF EDGE ME OF PLATE MP IS
       PART OF A DIFFRACTING WEDGE USED TO COMPUTE
       DIFFRACTED FIELDS FOR PLATE DIFFRACTED, CYLINDER
       REFLECTED RAY

COMMON LOGDIF  -----------------
THIS COMMON BLOCK CONTAINS INFORMATION THAT INDICATES WHETHER OR
NOT SLOPE AND CORNER DIFFRACTION
MECHANISMS ARE TO BE INCLUDED IN FIELD CALCULATIONS
LSLOPE  A LOGICAL VARIABLE USED TO INDICATE IF SLOPE DIFFRACTION
       IS DESIRED
              LSLOPE=T INDICATES SLOPE DIFFRACTION DESIRED
              LSLOPE=F INDICATES SLOPE DIFFRACTION NOT DESIRED
LCORNR A LOGICAL VARIABLE USED TO INDICATE IF CORNER DIFFRACTION
       IS DESIRED
              LCORNR=T INDICATES CORNER DIFFRACTION DESIRED
              LCORNR=F INDICATES CORNER DIFFRACTION NOT DESIRED

COMMON LPLCY  ----------------
THIS COMMON BLOCK CONTAINS LOGICAL VARIABLES INDICATING THE PRESENCE
OR ABSENCE OF PLATES AND CYLINDERS IN THE GEOMETRY (SPECIFIED IN
MAIN PROGRAM)
LPLA   A LOGICAL VARIABLE USED TO INDICATE THE PRESENCE OF AT
       LEAST ONE PLATE OR INFINITE GROUND PLATE
              LPLA=T INDICATES PLATES ARE PRESENT
              LPLA=F INDICATES PLATES NOT PRESENT
LCYL   A LOGICAL VARIABLE USED TO INDICATE THE PRESENCE OF
       AN ELLIPTIC CYLINDER
              LCYL=T INDICATES CYLINDER PRESENT
              LCYL=F INDICATES CYLINDER NOT PRESENT

COMMON LSHDP  ------------------------
THIS COMMON BLOCK IS USED TO TRANSFER DATA BETWEEN SUBROUTINE GEOM
AND SUBROUTINE PLAINT FOR THE TOTAL SHADOWING ALGORITHM
LSTS   A LOGICAL VARIABLE SET TRUE IF TOTAL SHADOWING ALGORITHM
       IS BEING USED
LSTD(14)  A LOGICAL ARRAY SUCH THAT
       LSTD(ML) IS SET TRUE IF PLATE ML TOTALLY SHADOWS PLATE MP
       FROM THE SOURCE

COMMON LSHDT  -------------------------
THIS COMMON BLOCK CONTAINS INFORMATION INDICATING PLATES THAT
ARE TOTALLY SHADOWED FROM THE SOURCE OR PLATES WHICH ARE SHADOWED
FROM OTHER PLATES (GENERATED IN SUB. GEOM AND USED IN MAIN PROGRAM)
LSHD(14)  A LOGICAL VARIABLE USED TO INDICATE IF PLATE MP IS TOTALLY
       SHADOWED FROM THE SOURCE BY ANY ONE PLATE OR THE CYLINDER
       LSHD(MP)=T INDICATES PLATE MP IS TOTALLY SHADOWED FROM
       DIRECT SOURCE RAYS
       LSHD(MP)=F INDICATES PLATE MP IS NOT TOTALLY SHADOWED
LIHD(14,14)  A LOGICAL VARIABLE USED TO INDICATE IF PLATES
       MP AND MPP CANNOT ILLUMINATE EACH OTHER
       LIHD(MP,MPP)=T INDICATES PLATES CANNOT ILLUMINATE EACH OTHER
       LIHD(MP,MPP)=F INDICATES PLATES CAN ILLUMINATE EACH OTHER

COMMON OUTPTD  ------------------
THIS COMMON BLOCK CONTAINS INFORMATION USED TO OBTAIN THE
PROPER FIELD OUTPUT IN SUBROUTINE OUTPUT.
LPRAD  THIS LOGICAL VARIABLE IS SET TRUE IF TOTAL POWER
       RADIATED BY THE SOURCES IS SPECIFIED BY THE USER
LRANG  THIS LOGICAL VARIABLE IS SET TRUE IF COMPUTED
       FAR-ZONE FIELD VALUES ARE TO INCLUDE RANGE FACTOR
       (CEXP(-J*R)/R)
PRAD   TOTAL POWER RADIATED (OR INPUT POWER) IN WATTS
       (SPECIFIED BY THE USER)
```

RANG THE DISTANCE FROM THE ORIGIN TO THE FAR FIELD
POINT IN METERS
WL THE WAVELENGTH IN METERS

COMMON PATDAT ------------------
THIS COMMON BLOCK DEFINES THE PATTERN CUT COORDINATE SYSTEM.
XPC(3) THIS DEFINES THE PATTERN CUT COORD SYSTEM X AXIS UNIT
VECTOR IN (XYZ) REF. COORD. SYS. COMPONENTS
THE X AXIS UNIT VECTOR IS GIVEN AS:

$\widehat{XPC}=\hat{X}*XPC(1)+\hat{Y}*XPC(2)+\hat{Z}*XPC(3)$

YPC(3) THIS DEFINES THE PATTERN CUT COORD SYS Y AXIS UNIT
VECTOR IN (XYZ) RCS COMPONENTS
THE Y AXIS UNIT VECTOR IS GIVEN AS:

$\widehat{YPC}=\hat{X}*YPC(1)+\hat{Y}*YPC(2)+\hat{Z}*YPC(3)$

ZPC(3) THIS DEFINES THE PATTERN CUT COORD SYS Z AXIS UNIT
VECTOR IN (XYZ) REF. COORD. SYS. COMPONENTS
THE Z AXIS UNIT VECTOR IS GIVEN AS:

$\widehat{ZPC}=\hat{X}*ZPC(1)+\hat{Y}*ZPC(2)+\hat{Z}*ZPC(3)$

COMMON PIS --------------------
THIS COMMON BLOCK CONTAINS MATHEMATICAL CONSTANTS BASED ON
THE NUMBER, PI WHICH ARE USED THROUGHOUT THE PROGRAM
THEY ARE DEFINED IN THE BLOCK DATA.

PI THE CONSTANT, PI (3.14159265)

TPI A CONSTANT, TWO TIMES PI (6.28318531)

DPR THE CONVERSION FACTOR FOR CONVERTING ANGULAR MEASUREMENTS
IN RADIANS TO DEGREES (=180/PI=57.2957795)

RPD THE CONVERSION FACTOR FOR CONVERTING ANGULAR MEASUREMENTS IN
DEGREES TO RADIANS (=PI/180=0.0174532925)

COMMON ROTRDT ----------------------
THIS COMMON BLOCK DEFINES THE NEW REFERENCE COORDINATE SYSTEM AXES
DIRECTIONS. IT IS DEFINED FROM INPUT DATA IN THE MAIN
PROGRAM AND IS USED IN SUBROUTINE ROTRAN TO TRANSFORM LOCATIONS
AND VECTORS FROM OLD REF COORD SYSTEM COMPONENTS TO NEW REFERENCE
COORDINATE SYSTEM COMPONENTS. THE NEW REFERENCE COORDINATE SYSTEM IS
THE CYLINDER COORDINATE SYSTEM (IF A CYLINDER IS PRESENT).
IF THE CYLINDER IS NOT PRESENT THE TRANSFORMATION IS NOT NECESSARY
SINCE THE REFERENCE COORDINATE SYSTEM REMAINS THE SAME COORDINATE
SYSTEM IN WHICH THE GEOMETRY WAS DEFINED

XCL(3) THIS DEFINES THE NEW REFERENCE COORDINATE SYSTEM X-AXIS UNIT
VECTOR IN OLD REFERENCE SYSTEM COMPONENTS
THE RCS X-AXIS UNIT VECTOR IS DEFINED AS:

$\hat{X}=\widehat{XO}*XCL(1)+\widehat{YO}*XCL(2)+\widehat{ZO}*XCL(3)$

YCL(3) THIS DEFINES THE NEW REFERENCE COORDINATE SYSTEM Y-AXIS UNIT
VECTOR IN OLD REFERENCE SYSTEM COMPONENTS
THE RCS Y-AXIS UNIT VECTOR IS DEFINED AS:

$\hat{Y}=\widehat{XO}*YCL(1)+\widehat{YO}*YCL(2)+\widehat{ZO}*YCL(3)$

ZCL(3) THIS DEFINES THE NEW REFERENCE COORDINATE SYSTEM Z-AXIS UNIT
VECTOR IN OLD REFERENCE SYSTEM COMPONENTS
THE RCS Z-AXIS UNIT VECTOR IS DEFINED AS:

$\hat{Z}=\widehat{XO}*ZCL(1)+\widehat{YO}*ZCL(2)+\widehat{ZO}*ZCL(3)$

WHERE $\widehat{XO},\widehat{YO},\widehat{ZO}$ ARE UNIT VECTORS OF THE OLD REFERENCE
COORD SYS AXES

COMMON SORINF ------------------
THIS COMMON BLOCK CONTAINS INFORMATION PERTAINING TO THE LOCATION
AND ORIENTATION OF THE SOURCE UNDER CONSIDERATION (SPECIFIED IN
MAIN PROGRAM)

XS(3) THE LOCATION OF THE SOURCE IN (XYZ) REFERENCE COORDINATE
SYSTEM COMPONENTS IN WAVELENGTHS

VXS(3,3) A 3X3 MATRIX DEFINING THE SOURCE COORDINATE
SYSTEM AXES UNIT VECTORS IN REFERENCE COORDINATE SYSTEM

```
        COMPONENTS:
                      X̂P=X̂*VXS(1,1)+Ŷ*VXS(1,2)+Ẑ*VXS(1,3)
                      ŶP=X̂*VXS(2,1)+Ŷ*VXS(2,2)+Ẑ*VXS(2,3)
                      ẐP=X̂*VXS(3,1)+Ŷ*VXS(3,2)+Ẑ*VXS(3,3)

COMMON SOURSF  -------------------
THIS COMMON BLOCK CONTAINS A SOURCE FIELD FACTOR.
IT IS COMPUTED IN SUBROUTINES GEOM AND GEOMC AND IS USED
IN SUBROUTINE SOURCE AND SOURCP.
FACTOR  THIS IS A COEFFICIENT OF THE SOURCE FIELD USED
        TO OBTAIN THE CORRECT FIELD MAGNITUDE FOR SOURCES
        MOUNTED ON PLATES OR END CAPS (IN ORDER TO
        COMPENSATE FOR IMAGE EFFECTS).  FACTOR IS GIVEN AS
        FOLLOWS:
             FOR ELECTRIC SOURCES:
                 FOR SOURCE NOT MOUNTED ON PLATE OR END CAP,
                    FACTOR=1.Ø
                 FOR SOURCE MOUNTED NORMAL TO PLATE OR END CAP,
                    FACTOR=1.Ø
                 FOR SOURCE MOUNTED ON PLATE OR END CAP BUT NOT
                 NORMAL TO IT,
                    FACTOR=Ø.5
             FOR MAGNETIC SOURCES:
                 FOR SOURCE NOT MOUNTED ON PLATE OR END CAP,
                    FACTOR=1.Ø
                 FOR SOURCE MOUNTED ON PLATE OR END CAP AND
                 PARALLEL TO IT,
                    FACTOR=2.Ø
                 FOR SOURCE MOUNTED ON PLATE OR END CAP, BUT NOT
                 PARALLEL TO IT,
                    FACTOR=1.Ø

COMMON SRFACC  -----------------------
THIS COMMON BLOCK IS DEFINED IN SUBROUTINE GEOMC AND IS USED IN THE
MAIN PROGRAM
LSRFC(MC)  A LOGICAL VARIABLE INDICATING WHETHER OR NOT
        THE SOURCE UNDER CONSIDERATION IS MOUNTED ON CYLINDER
        END CAP MC
        LSRFC(MC)=T INDICATES SOURCE MOUNTED ON END CAP MC
        LSRFC(MC)=F INDICATES SOURCE NOT MOUNTED ON END CAP MC

COMMON SURFAC  ------------------
THIS BLOCK IS DEFINED IN SUBROUTINE GEOM AND IS USED IN THE MAIN
PROGRAM AND IN SEVERAL SUBROUTINES
LSURF(14)  A LOGICAL VARIABLE INDICATING WHETHER OR NOT
        THE SOURCE UNDER CONSIDERATION IS MOUNTED ON PLATE MP
        LSURF(MP)=T INDICATES SOURCE MOUNTED ON PLATE MP
        LSURF(MP)=F INDICATES SOURCE NOT MOUNTED ON PLATE MP


COMMON TEST  --------------------
THIS COMMON BLOCK CONTAINS LOGICAL VARIABLES USED TO INSTRUCT
THE CODE WHETHER OR NOT A PRINT-OUT OF TEST DATA IS DESIRED.
LDEBUG  THIS LOGICAL VARIABLE IS SET TRUE IF DEBUGGING DATA IS TO
        BE PRINTED ON LINE PRINTER
LTEST   THIS LOGICAL VARIABLE IS SET TRUE IF TEST DATA IS TO

        BE PRINTED ON LINE PRINTER

COMMON THPHUV  ------------------
THIS COMMON BLOCK CONTAINS INFORMATION DEFINING THE THETA AND PHI
POLARIZATION UNIT VECTORS FOR THE OBSERVATION DIRECTION (COMPUTED IN
MAIN PROGRAM)
DT(3)   THE THETA UNIT VECTOR FOR OBSERVATION DIRECTION D
        IN RCS COMPONENTS:
               D̂T=X̂*DT(1)+Ŷ*DT(2)+Ẑ*DT(3)
DP(2)   THE PHI UNIT VECTOR FOR OBSERVATION DIRECTION D IN REFERENCE
```

```
          COORDINATE SYSTEM COMPONENTS:
                 DP=X̂*DP(1)+Ŷ*DP(2)+Ẑ*DP(3)

COMMON TOPD --------------------
THIS COMMON BLOCK CONTAINS A CONSTANT USED IN THE DIFFRACTION
SUBROUTINES
TOP      THE COMPLEX CONSTANT, -CEXP(-J*PI/4)
```

# CHAPTER VI
## SYSTEM LIBRARY FUNCTIONS USED BY CODE

| | | |
|---|---|---|
| ACOS(X) | = | arccos of X; result in radians |
| AINT(X) | = | X truncated to an integer and set real |
| ALOG10(X) | = | log to base ten of X |
| ATAN2(Y,X) | = | arctangent of Y/X; result in radians covering all four quadrants |
| CABS(Z) | = | magnitude of the complex number, Z |
| CEXP(Z) | = | complex exponential ($e^Z$) |
| CLOG(Z) | = | complex log of Z ($\ln Z + j \tan^{-1} \frac{Im(Z)}{Re(Z)}$) |
| CONJG(X) | = | complex conjugate of Z |
| COS(X) | = | cosine of X |
| CSQRT(Z) | = | square root of a complex number, Z |
| SIGN(X,Y) | = | sign of Y times absolute value of X |
| SIN(X) | = | sine of X |
| SQRT(X) | = | square root of X |
| TAN(X) | = | tangent of X |

# REFERENCES

1. R.J. Marhefka, "Analysis of Aircraft Wing-Mounted Antenna Patterns," Report 2902-25, June 1976, The Ohio State University ElectroScience Laboratory, Department of Electrical Engineering; prepared under Grant No. NGL 36-008-138 for National Aeronautics and Space Administration.

2. W.D. Burnside, M.C. Gilreath, R.J. Marhefka, and C.L. Yu, "A Study of KC-135 Aircraft Antenna Patterns," IEEE Trans. on Antennas and Propagation, Vol. AP-23, No. 3, May 1975, pp. 309-316.

3. C.L. Yu, W.D. Burnside and M.C. Gilreath, "Volumetric Pattern Analysis of Airborne Antennas," IEEE Trans. on Antennas and Propagation, Vol. AP-26, No. 5, Sept. 1978, pp. 636-641.

4. R.G. Kouyoumjian and P.H. Pathak, "A Uniform Geometrical Theory of Diffraction for an Edge in a Perfectly-Conducting Surface," Proc. IEEE, Vol. 62, November 1974, pp. 1448-1461.

5. R.G. Kouyoumjian, "The Geometrical Theory of Diffraction and Its Applications," Numerical and Asymptotic Techniques in Electromagnetics, edited by R. Mittra, Spring-Verlag, New York, 1975.

6. P.H. Pathak, W.D. Burnside and R.J. Marhefka, "A Uniform GTD Analysis of the Diffraction of Electromagnetic Waves by a Smooth Convex Surface," submitted for publication to IEEE Trans. on Antennas and Propagation. (Also Report 784583-4, April 1979, The Ohio State University ElectroScience Laboratory, Department of Electrical Engineering; prepared under Contract No. N62269-76-C-0554 for Naval Air Development Center.

7. G.J. Burke and A.J. Poggio, "Numerical Electromagnetic Code (NEC) - Method of Moments," NOSC/TD 116, July 1977, Naval Ocean Systems Center, San Diego, California 92152.

8. R.J. Marhefka and W.D. Burnside, "NEC - Basic Scattering Code User's Manual," Report 784508-18, in progress, The Ohio State University ElectroScience Laboratory, Department of Electrical Engineering; prepared under Contract No. N00123-76-C-1371 for Naval Regional Procurement Office.

9. W.D. Burnside and P.H. Pathak, "A Corner Diffraction Coefficient," to appear.

10. Y.M. Hwang and R.G. Kouyoumjian, "A Dyadic Diffraction Coefficient for an Electromagnetic Wave which is Rapidly Varying at an Edge," USNC-URSI 1974 Annual Meeting, Boulder, CO., Oct. 1974.

11. IBM, System 1360 Scientific Subroutine Package, (360A-CM-03Y), Version III, Programmer's Manual, IBM Corp., 1968, p. 303.

12. E.D. Greer and W.D. Burnside, "High Frequency Near Field Scattering by an Elliptic Disk," Report 4583-1, December 1976, The Ohio State University ElectroScience Laboratory, Department of Electrical Engineering; prepared under Contract No. N62269-76-C-0554 for Naval Air Development Center.

13. J. Boersma, "Computation of Fresnel Integrals,""Math Comp., Vol. 14, 1960, p. 380.

14. P.H. Pathak, "An Asymptotic Analysis of the Scattering of Plane Waves by a Smooth Convex Cylinder," paper to appear in J. Radio Science. (Also The Ohio State University ElectroScience Laboratory Tech. Report 784583-3, March 1978).

15. N.A. Logan, "General Research in Diffraction Theory," Vol. I, LMSD-288087; and Vol. II, LMSD-288088, Missiles and Space Division, Lockheed Aircraft Corp., 1959.

16. A. Ralston, A First Course in Numerical Analysis, McGraw-Hill Book Co., New York, 1965, pp. 371-373.